Journal of Computational Physics 326 (2016) 516–543

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems

M.J. Zahr [a,*], P.-O. Persson [b]

[a] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94035, United States
[b] Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720-3840, United States

ARTICLE INFO

*Article history:*
Received 14 December 2015
Received in revised form 5 September 2016
Accepted 7 September 2016
Available online 12 September 2016

*Keywords:*
Fully discrete adjoint method
Discontinuous Galerkin methods
Deforming domain conservation law
Optimal control
High-order methods
PDE-constrained optimization

ABSTRACT

The fully discrete adjoint equations and the corresponding adjoint method are derived for a globally high-order accurate discretization of conservation laws on parametrized, deforming domains. The conservation law on the deforming domain is transformed into one on a fixed reference domain by the introduction of a time-dependent mapping that encapsulates the domain deformation and parametrization, resulting in an Arbitrary Lagrangian–Eulerian form of the governing equations. A high-order discontinuous Galerkin method is used to discretize the transformed equation in space and a high-order diagonally implicit Runge–Kutta scheme is used for the temporal discretization. Quantities of interest that take the form of space–time integrals are discretized in a solver-consistent manner. The corresponding fully discrete adjoint method is used to compute *exact* gradients of quantities of interest along the manifold of solutions of the fully discrete conservation law. These quantities of interest and their gradients are used in the context of gradient-based PDE-constrained optimization.

The adjoint method is used to solve two optimal shape and control problems governed by the isentropic, compressible Navier–Stokes equations. The first optimization problem seeks the energetically optimal trajectory of a 2D airfoil given a required initial and final spatial position. The optimization solver, driven by gradients computed via the adjoint method, reduced the total energy required to complete the specified mission nearly an order of magnitude. The second optimization problem seeks the energetically optimal flapping motion and time-morphed geometry of a 2D airfoil given an equality constraint on the $x$-directed impulse generated on the airfoil. The optimization solver satisfied the impulse constraint to greater than 8 digits of accuracy and reduced the required energy between a factor of 2 and 10, depending on the value of the impulse constraint, as compared to the nominal configuration.

## 1. Introduction

Optimization problems constrained by Partial Differential Equations (PDEs) commonly arise in engineering practice, particularly in the context of design or control of physics-based systems. A majority of the research in PDE-constrained

[*] Corresponding author.
*E-mail addresses:* mzahr@stanford.edu (M.J. Zahr), persson@berkeley.edu (P.-O. Persson).

http://dx.doi.org/10.1016/j.jcp.2016.09.012
0021-9991/© 2016 Elsevier Inc. All rights reserved.

optimization has been focused on *steady* or *static* PDEs, with a large body of literature detailing many aspects of the subject, including continuous and discrete adjoint methods [1–5], parallel implementations [6,7], one-shot or infeasible path methods [6,8], and generalized reduced gradient or feasible path methods [1,9]. This emphasis on steady problems is largely due to the fact that (a) static analysis is sufficient for a large class of problems of interest and (b) unsteady analysis is expensive to perform in a many-query setting, such as optimization [10]. However, there is a large class of problems where steady analysis is insufficient, such as problems that are inherently dynamic and problems where a steady-state solution does not exist or cannot be found reliably with numerical methods. Flapping flight is an example of the first type, a fundamentally unsteady problem that has become increasingly relevant due to its application to Micro-Aerial Vehicles (MAVs) [11]. Systems with chaotic solutions, such as those encountered in turbulent flows, are an example of the second type of problems where steady analysis breaks down. Design and control of these types of systems calls for *time-dependent* PDE-constrained optimization of the form

$$\begin{aligned}
&\underset{\boldsymbol{U},\,\boldsymbol{\mu}}{\text{minimize}} && \int_0^T \int_{\boldsymbol{\Gamma}} j(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt \\
&\text{subject to} && \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{c}(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt \leq 0 \\
& && \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0
\end{aligned} \tag{1}$$

where the last constraint corresponds to a system of conservation laws with solution $\boldsymbol{U}$, parametrized by $\boldsymbol{\mu}$; the objective and constraint functions of the optimization take the form of space–time integrals of pointwise, instantaneous quantities of interest $j$ and $\boldsymbol{c}$ over the surface of the body $\boldsymbol{\Gamma}$.

In this work, the large computational cost associated with time-dependent PDE-constrained optimization will be addressed by two means. The first is the development of a *globally high-order* numerical discretization of conservation laws on deforming domains.[1] For many important problems, high-order methods have been shown to require fewer spatial degrees of freedom [12] and time steps [13,14] for a given level of accuracy compared to low-order methods. Highly accurate quantities of interest, usually the time-average of a relevant surface- or volume-integrated quantity, is paramount, at least at convergence, since they drive the optimization trajectory through the objective function and constraints. Large errors in quantities of interest will cause the optimization procedure to be driven by discretization errors causing termination at a suboptimal design/control. The second approach to reduce the computational impact of time-dependent optimization is the use of gradient-based optimization techniques due to their rapid convergence properties, particularly when compared to derivative-free alternatives.

An efficient technique for computing derivatives of optimization functionals, required by gradient-based optimization solvers, is the *adjoint method*. It has proven its utility in the context of output-based mesh adaptivity and gradient-based PDE-constrained optimization as only a single linearized dual solve is required to compute the gradient of a single quantity of interest with respect to any number of parameters. In the context of partial differential equations, the adjoint equations can be derived at either the continuous, semi-discrete, or fully discrete level. The fully discrete adjoint method will be the focus of this work as it ensures discrete consistency [5,10] of computed gradients, i.e. the gradient of the discrete solution, including discretization errors, is computed. Discrete consistency is beneficial in the context of gradient-based optimization as inconsistent gradients may cause convergence of black-box optimizers to be slowed or hindered [15], unless specialized optimization algorithms are employed that handle gradient inexactness [16].

In this work, a *globally high-order* numerical discretization of general systems of conservation laws, defined on deforming domains, is introduced and the corresponding fully discrete adjoint equations derived. The goal is to harness the advantages of high-order methods in the context of *gradient-based* optimization. The solution of the adjoint equations – the dual solution – will be used to construct exact gradients of fully discrete quantities of interest. A Discontinuous Galerkin Arbitrary Lagrangian–Eulerian (DG-ALE) method [17] is used for the high-order spatial discretization. Previous work on the adjoint method for conservation laws on deforming domains predominantly considers a Finite Volume (FV) spatial discretization [1–4,18,13,5,19], and recently extended to DG-ALE schemes [20–22]. The DG-ALE discretization is chosen rather than FV due to its stable, high-order discretization of convective fluxes. The Geometric Conservation Law (GCL) is satisfied in the DG-ALE scheme through the introduction of an element-level auxiliary equation. The fully discrete adjoint equations derived in this work fully incorporate this GCL augmentation [23], ensuring discrete consistency is maintained.

High-order temporal discretization will be achieved using a Diagonally Implicit Runge–Kutta (DIRK) [24] method, marking a departure from previous work on unsteady adjoints, which has mostly considered temporal discretization via Backward Differentiation Formulas (BDF) [9,18,13,19,22,25], with some work on space–time DG discretizations [23]. Apart from being limited to second-order accuracy, if A-stability is required, high-order BDF schemes require special techniques for initialization [14]. While DIRK schemes require additional work to achieve high-order convergence in the form of additional nonlinear

[1] As with all works on high-order methods, high-order accuracy relies on sufficient regularity in the solution.

solves at a given timestep, this investment has been shown to be worthwhile [14]. The fully discrete, time-dependent adjoint equations corresponding to a Runge–Kutta temporal discretization has been studied in the context of ODEs [26] and optimal control [27,28]. A difficulty that arises when considering Runge–Kutta discretizations is the stages encountered during the primal (forward) and dual (reverse) solves will not align if the continuous or semi-discrete adjoint method is employed. This difficulty cannot arise in the fully discrete formulation as only the terms computed in the primal solve are used in the derivation of the discrete adjoint equations.

High-order discretization of integrated quantities of interest will be done in a solver-consistent manner, that is, spatial integrals will be evaluated via integration of the finite element shape functions used for the spatial discretization and temporal integrals evaluated via the DIRK scheme from the temporal discretization. This ensures the discretization order of quantities of interest exactly matches the PDE discretization. This marks a departure from the traditional method of simply using the trapezoidal rule for temporal integration of quantities of interest [10,13,19,22], which is limited to second-order accuracy. Since the fully discrete adjoint method is used, discretization of the quantities of interest is accounted for in the adjoint equations, which is the final component in ensuring discrete consistency of their gradients.

This notion of a solver-consistent discretization of quantities of interest has an immediate connection to adjoint-consistency for stationary problems [29,30]. A necessary condition for adjoint-consistency is that the quantity of interest is discretized with the same scheme as the governing equations and boundary conditions [31]; however, this does not refer to the practical issue of evaluating the integrals that arise in the finite-dimensional primal or adjoint residual. Solver-consistency ensures the same approximation is used to evaluate the integrals arising the primal residual and quantity of interest, and therefore the adjoint residual. A key property of adjoint-consistent discretizations is they possess optimal convergence rates in the $L^2$-norm and in quantities of interest [32–34].

With the high-order numerical discretization in place, and the corresponding adjoint method developed, the proposed methodology is employed to solve deforming-domain PDE-constrained optimization problems, such as optimal control and shape optimization for fluid flows. Existing techniques for parametrizing the domain deformation include (a) boundary-driven deformation where the deformation of a boundary is parametrized using existing techniques [35,36] and the domain is deformed using a structural analogy [37,13,19], and (b) analytical expressions for direct domain deformation [17,21]. The latter approach is adopted in this work. The proposed methodology, including the high-order primal discretization, fully discrete adjoint method, and domain parametrization, will be demonstrated on relevant optimal control and shape optimization problems. An important component will be the incorporation of *solution-based* constraints, which has previously been done only through heuristic penalization of the objective function [22].

The remainder of this document is organized as follows. Section 2 introduces the general form of a system of conservation laws defined on deforming domains, as well as the globally high-order discretization using DG-in-space, DIRK-in-time and a solver-consistent discretization of output integrals. Section 3 derives the fully discrete adjoint method corresponding to the high-order discretization from Section 2, with relevant implementational details discussed in Section 4. Section 5 couples the adjoint framework to state-of-the-art optimization software to solve a realistic optimization problem of determining energetically optimal flapping motions of an airfoil. An alternate derivation of the adjoint equations is provided in Appendix A by interpreting the dual variables as Lagrange multipliers of an auxiliary PDE-constrained optimization problem.

## 2. Governing equations and discretization

This section is devoted to the treatment of conservation laws on a *parametrized, deforming domain* using an Arbitrary Lagrangian–Eulerian (ALE) description of the governing equations and exposition of a globally high-order numerical discretization of the ALE form of the system of conservation laws. Subsequently, Section 3 will develop the corresponding fully discrete adjoint equations and the adjoint method for constructing gradients of quantities of interest.

The methods introduced in this work are not necessarily limited to Partial Differential Equations (PDE) that can be written as conservation laws (2). In Section 2.2, the chosen spatial discretization (discontinuous Galerkin Arbitrary Lagrangian–Eulerian method) is applied to the PDE, resulting in a system of first-order Ordinary Differential Equations (ODE), which is the point of departure for all adjoint-related derivations. Time-dependent PDEs that are not conservation laws can be written similarly at the semi-discrete level after application of an appropriate spatial discretization, e.g. a continuous finite element method for parabolic PDEs. In this work, the scope is limited to first-order temporal systems, or those which are recast as such.

### 2.1. System of conservation laws on deforming domain: arbitrary Lagrangian–Eulerian description

Consider a general system of conservation laws, defined on a parametrized, deforming domain, $v(\boldsymbol{\mu}, t)$, written at the continuous level as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = 0 \quad \text{in} \ \ v(\boldsymbol{\mu}, t) \tag{2}$$

where the physical flux is decomposed into an inviscid and a viscous part $\boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = \boldsymbol{F}^{inv}(\boldsymbol{U}) + \boldsymbol{F}^{vis}(\boldsymbol{U},\ \nabla \boldsymbol{U})$, $\boldsymbol{U}(\boldsymbol{x}, \boldsymbol{\mu}, t)$ is the solution of the system of conservation laws, $t \in (0, T)$ represents time, and $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$ is a vector of parameters. This

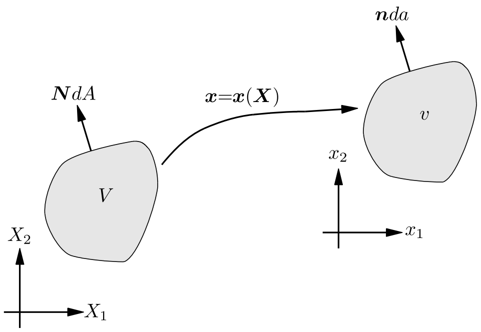

**Fig. 1.** Time-dependent mapping between reference and physical domains.

work will focus on the case where the *domain* is parametrized by $\boldsymbol{\mu}$, although extension to other types of parameters, e.g. constants defining the conservation law, is straightforward.

The conservation law on a deforming domain is transformed into a conservation law on a *fixed* reference domain through the introduction of a time-dependent mapping between the physical and reference domains, resulting in an Arbitrary Lagrangian–Eulerian description of the governing equations.

Denote the physical domain by $v(\boldsymbol{\mu},t) \subset \mathbb{R}^{n_{sd}}$ and the fixed, reference domain by $V \subset \mathbb{R}^{n_{sd}}$, where $n_{sd}$ is the number of spatial dimensions. At each time $t$, let $\mathcal{G}$ be a time-dependent diffeomorphism between the reference domain and physical domain: $\boldsymbol{x}(\boldsymbol{X},\boldsymbol{\mu},t) = \mathcal{G}(\boldsymbol{X},\boldsymbol{\mu},t)$, where $\boldsymbol{X} \in V$ is a point in the reference domain and $\boldsymbol{x}(\boldsymbol{X},\boldsymbol{\mu},t) \in v(\boldsymbol{\mu},t)$ is the corresponding point in the physical domain at time $t$ and parameter configuration $\boldsymbol{\mu}$ (see Fig. 1).

The transformed system of conservation laws from (2), under the mapping $\mathcal{G}$, defined on the reference domain takes the form

$$\left.\frac{\partial \boldsymbol{U_X}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F_X}(\boldsymbol{U_X},\ \nabla_{\boldsymbol{X}}\boldsymbol{U_X}) = 0 \tag{3}$$

where $\nabla_{\boldsymbol{X}}$ denotes spatial derivatives with respect to the reference variables, $\boldsymbol{X}$. The transformed state vector, $\boldsymbol{U_X}$, and its corresponding spatial gradient with respect to the reference configuration take the form

$$\boldsymbol{U_X} = g\boldsymbol{U}, \qquad \nabla_{\boldsymbol{X}}\boldsymbol{U_X} = g^{-1}\boldsymbol{U_X}\frac{\partial g}{\partial \boldsymbol{X}} + g\nabla\boldsymbol{U}\cdot\boldsymbol{G}, \tag{4}$$

where $\boldsymbol{G} = \nabla_{\boldsymbol{X}}\mathcal{G}$, $g = \det(\boldsymbol{G})$, $\boldsymbol{v_G} = \frac{\partial \boldsymbol{x}}{\partial t} = \frac{\partial \mathcal{G}}{\partial t}$, and the arguments have been dropped, for brevity. The transformed fluxes are

$$\begin{aligned} \boldsymbol{F_X}(\boldsymbol{U_X},\nabla_{\boldsymbol{X}}\boldsymbol{U_X}) &= \boldsymbol{F}^{inv}_{\boldsymbol{X}}(\boldsymbol{U_X}) + \boldsymbol{F}^{vis}_{\boldsymbol{X}}(\boldsymbol{U_X},\nabla_{\boldsymbol{X}}\boldsymbol{U_X}), \\ \boldsymbol{F}^{inv}_{\boldsymbol{X}}(\boldsymbol{U_X}) &= g\boldsymbol{F}^{inv}(g^{-1}\boldsymbol{U_X})\boldsymbol{G}^{-T} - \boldsymbol{U_X}\otimes\boldsymbol{G}^{-1}\boldsymbol{v_G}, \\ \boldsymbol{F}^{vis}_{\boldsymbol{X}}(\boldsymbol{U_X},\nabla_{\boldsymbol{X}}\boldsymbol{U_X}) &= g\boldsymbol{F}^{vis}\left(g^{-1}\boldsymbol{U_X}, g^{-1}\left[\nabla_{\boldsymbol{X}}\boldsymbol{U_X} - g^{-1}\boldsymbol{U_X}\frac{\partial g}{\partial \boldsymbol{X}}\right]\boldsymbol{G}^{-1}\right)\boldsymbol{G}^{-T}. \end{aligned} \tag{5}$$

For details regarding the derivation of the transformed equations, the reader is referred to [17].

When integrated using inexact numerical schemes, this ALE formulation does not satisfy the Geometric Conservation Law (GCL) [38,17]. This is overcome by introduction of an auxiliary variable $\bar{g}$, defined as the solution of

$$\frac{\partial \bar{g}}{\partial t} - \nabla_{\boldsymbol{X}}\cdot\left(g\boldsymbol{G}^{-1}\boldsymbol{v_G}\right) = 0. \tag{6}$$

The auxiliary variable, $\bar{g}$ is used to modify the *transformed* conservation law according to

$$\left.\frac{\partial \boldsymbol{U}_{\bar{\boldsymbol{X}}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}},\ \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}}) = 0 \tag{7}$$

where the GCL-transformed state variables are

$$\boldsymbol{U}_{\bar{\boldsymbol{X}}} = \bar{g}\boldsymbol{U}, \qquad \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}} = \bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}\frac{\partial \bar{g}}{\partial \boldsymbol{X}} + \bar{g}\nabla\boldsymbol{U}\cdot\boldsymbol{G} \tag{8}$$

and the corresponding fluxes

$$\begin{aligned}\boldsymbol{F}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= \boldsymbol{F}^{inv}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}) + \boldsymbol{F}^{vis}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\bar{\boldsymbol{X}}}),\\ \boldsymbol{F}^{inv}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= g\boldsymbol{F}^{inv}(\bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}})\boldsymbol{G}^{-T} - \boldsymbol{U}_{\bar{\boldsymbol{X}}} \otimes \boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}},\\ \boldsymbol{F}^{vis}_{\bar{\boldsymbol{X}}}(\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\bar{\boldsymbol{X}}}) &= g\boldsymbol{F}^{vis}\left(\bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}, \bar{g}^{-1}\left[\nabla_{\boldsymbol{X}}\boldsymbol{U}_{\bar{\boldsymbol{X}}} - \bar{g}^{-1}\boldsymbol{U}_{\bar{\boldsymbol{X}}}\frac{\partial \bar{g}}{\partial \boldsymbol{X}}\right]\boldsymbol{G}^{-1}\right)\boldsymbol{G}^{-T}.\end{aligned} \tag{9}$$

It was shown in [17] that the transformed equations (7) satisfy the GCL. In the next section, the ALE description of the governing equations (3) and (7) will be converted to first-order form and discretized via a high-order discontinuous Galerkin method.

### 2.2. Spatial discretization: arbitrary Lagrangian–Eulerian discontinuous Galerkin method

The ALE description of the conservation law without GCL augmentation will be considered first. To proceed, the second-order system of partial differential equations in (3) is converted to first-order form

$$\begin{aligned}\left.\frac{\partial \boldsymbol{U}_{\boldsymbol{X}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}},\ \boldsymbol{Q}_{\boldsymbol{X}}) &= 0\\ \boldsymbol{Q}_{\boldsymbol{X}} - \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}} &= 0,\end{aligned} \tag{10}$$

where $\boldsymbol{Q}_{\boldsymbol{X}}$ is introduced as an auxiliary variable to represent the spatial gradient of the $\boldsymbol{U}_{\boldsymbol{X}}$. Equation (10) is discretized using a standard nodal discontinuous Galerkin finite element method [39], which after local elimination of the auxiliary variables $Q_{\boldsymbol{X}}$ leads to the following system of ODEs

$$\mathbb{M}_{\boldsymbol{X}}\frac{\partial \boldsymbol{u}_{\boldsymbol{X}}}{\partial t} = \boldsymbol{r}_{\boldsymbol{u}_{\boldsymbol{X}}}(\boldsymbol{u}_{\boldsymbol{X}}, \boldsymbol{\mu}, t), \tag{11}$$

where $\mathbb{M}_{\boldsymbol{X}}$ is the block-diagonal, symmetric, *fixed* mass matrix, $\boldsymbol{u}_{\boldsymbol{X}}$ is the vectorization of $\boldsymbol{U}_{\boldsymbol{X}}$ at all nodes in the high-order mesh, and $\boldsymbol{r}_{\boldsymbol{u}_{\boldsymbol{X}}}$ is the nonlinear function defining the DG discretization of the inviscid and viscous fluxes.

The GCL augmentation is treated identically, i.e. conversion to first-order form and subsequent application of the discontinuous Galerkin finite element method, where $\boldsymbol{U}_{\bar{\boldsymbol{X}}}$ is taken as the state variable. The result is a system of ODEs corresponding to a high-order ALE scheme that satisfies the GCL

$$\begin{aligned}\mathbb{M}_{\bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial t} &= \boldsymbol{r}_{\bar{\mathbf{g}}}(\boldsymbol{\mu}, t)\\ \mathbb{M}_{\boldsymbol{X}}\frac{\partial \boldsymbol{u}_{\bar{\boldsymbol{X}}}}{\partial t} &= \boldsymbol{r}_{\boldsymbol{u}_{\bar{\boldsymbol{X}}}}(\boldsymbol{u}_{\bar{\boldsymbol{X}}}, \bar{\mathbf{g}}, \boldsymbol{\mu}, t)\end{aligned} \tag{12}$$

where each term is defined according to their counterparts in (11). From the conservation law defining $\bar{g}$ (6), the corresponding flux is continuous, implying the physical flux $g\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}}$ can be used as the numerical flux. This implies no information is required from neighboring elements and (6) can be solved at the element level, i.e. statically condensed. Furthermore, the $\bar{\mathbf{g}}$ residual, $\boldsymbol{r}_{\bar{\mathbf{g}}}$, does not depend on $\bar{\mathbf{g}}$ itself since the physical flux $g\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{G}}$ is independent of $\bar{g}$.

Since the equation for $\bar{\mathbf{g}}$ does not depend on $\boldsymbol{u}_{\bar{\boldsymbol{X}}}$, it can be solved independently of the equation for $\boldsymbol{u}_{\bar{\boldsymbol{X}}}$. This enables $\bar{\mathbf{g}}$ to be considered an implicit function of $\boldsymbol{\mu}$, i.e. $\bar{\mathbf{g}} = \bar{\mathbf{g}}(\boldsymbol{\mu}, t)$ through application of the implicit function theorem. Then, (12) reduces to

$$\mathbb{M}_{\boldsymbol{X}}\frac{\partial \boldsymbol{u}_{\bar{\boldsymbol{X}}}}{\partial t} = \boldsymbol{r}_{\boldsymbol{u}_{\bar{\boldsymbol{X}}}}(\boldsymbol{u}_{\bar{\boldsymbol{X}}}, \bar{\mathbf{g}}(\boldsymbol{\mu}, t), \boldsymbol{\mu}, t). \tag{13}$$

Equations (11) and (13) are abstracted into the following system of ODEs

$$\mathbb{M}\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t), \tag{14}$$

for convenience in the derivation of the fully discrete adjoint equations. Evaluation of the residual, $\boldsymbol{r}$, in (14) at parameter $\boldsymbol{\mu}$ and time $t$ requires evaluation of the mapping, $\boldsymbol{x}(\boldsymbol{\mu}, t)$ and $\dot{\boldsymbol{x}}(\boldsymbol{\mu}, t)$, and $\bar{\mathbf{g}}(\boldsymbol{\mu}, t)$, if GCL augmentation is employed. The implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$ requires special treatment when computing derivatives with respect to $\boldsymbol{\mu}$, which will be required in the adjoint method (Section 3). Discussion of treatment of such terms will be deferred to Section 4.1.2.

A convenient property of this DG-ALE scheme is that all computations are performed on the reference domain which is *independent* of time and parameter. An implication of this is that the mass matrix of the ODE (14) is also time- and parameter-independent. This property simplifies all adjoint computations introduced in Section 3 as terms involving $\dfrac{\partial \mathbb{M}}{\partial \boldsymbol{u}}$ and $\dfrac{\partial \mathbb{M}}{\partial \boldsymbol{\mu}}$ are identically zero. This, in turn, simplifies the implementation of the adjoint method and translates to computational savings since contractions with these third-order tensor are not required; see [9] for a discretization with

**Table 1**
Butcher tableau for $s$-stage diagonally implicit Runge–Kutta scheme.

| | | | | |
|---|---|---|---|---|
| $c_1$ | $a_{11}$ | | | |
| $c_2$ | $a_{21}$ | $a_{22}$ | | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | |
| $c_s$ | $a_{s1}$ | $a_{s2}$ | $\cdots$ | $a_{ss}$ |
| | $b_1$ | $b_2$ | $\cdots$ | $b_s$ |

parameter-dependent mass matrices and the corresponding adjoint derivation. In subsequent sections, it will be assumed that the mass matrix is time- and parameter-independent.

The DG-ALE scheme outlined in this section constitutes a *spatial* discretization, which yields a system of Ordinary Differential Equations (ODEs) when applied to the PDE in (2). The semi-discrete form of the conservation law is the point of departure for the remainder of this document. The subsequent development applies to any system of ODEs of the form (14) without relying on the specific spatial discretization scheme employed. The DG-ALE scheme was chosen to provide a high-order, stable spatial discretization of the conservation law (2). With the high-order spatial discretization of the state equations introduced, focus is shifted toward high-order temporal discretization to yield a globally high-order accurate numerical scheme.

### 2.3. Temporal discretization: diagonally implicit Runge–Kutta

The two prevailing classes of high-order implicit temporal integration schemes are: (a) Backward Differentiation Formulas (BDF) and (b) Implicit Runge–Kutta (IRK). BDF schemes are popular since high-order accuracy can be achieved at the cost of the solution of a single nonlinear system of equations of size $N_{\boldsymbol{u}}$ at each time step. However, they suffer from initialization issues and are limited to second-order, if A-stability is required. In contrast, IRK schemes are single-step methods that can be A-stable and arbitrarily high-order, at the cost of solving an *enlarged* nonlinear system of equations of size $s \cdot N_{\boldsymbol{u}}$, for an $s$-stage scheme. For practical problems, this can be prohibitively expensive, in terms of memory and CPU time.

A particular subclass of the IRK schemes, known as Diagonally Implicit Runge–Kutta (DIRK) schemes [24], are capable of achieving high-order accuracy with the desired stability properties, without requiring the solution of an enlarged system of equations. The DIRK schemes are defined by a *lower triangular* Butcher tableau (Table 1) and take the following form when applied to (14)

$$\begin{aligned}
\boldsymbol{u}^{(0)} &= \boldsymbol{u}_0(\boldsymbol{\mu}) \\
\boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
\mathbb{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right),
\end{aligned} \tag{15}$$

for $n = 1, \dots, N_t$ and $i = 1, \dots, s$, where $N_t$ are the number of time steps in the temporal discretization and $s$ is the number of stages in the DIRK scheme. The temporal domain, $[0,\ T]$ is discretized into $N_t$ segments with endpoints $\{t_0, t_1, \dots, t_{N_t}\}$, with the $n$th segment having length $\Delta t_n = t_n - t_{n-1}$ for $n = 1, \dots, N_t$. Additionally, in (15), $\boldsymbol{u}_i^{(n)}$ is used to denote the approximation of $\boldsymbol{u}^{(n)}$ at the $i$th stage of time step $n$

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}_i^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{k}_1^{(n)}, \dots,\ \boldsymbol{k}_s^{(n)}) = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)}. \tag{16}$$

From (15), a complete time step requires the solution of a sequence of $s$ nonlinear systems of equation of size $N_{\boldsymbol{u}}$.

Given this exposition on the spatio-temporal discretization of the deforming domain conservation law, the expression for the quantity of interest must be discretized. The next section introduces a spatio-temporal discretization of the quantities of interest that is *consistent* with that of the conservation law itself.

### 2.4. Solver-consistent discretization of quantities of interest

In this section, the high-order discretization of quantities of interest – space–time integrals of a nonlinear function of the solution of the conservation law – is considered. The output quantity takes the form

$$\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}, t) = \int_0^t \int_{\boldsymbol{\Gamma}} w(\boldsymbol{x}, \tau) f(\boldsymbol{U}, \boldsymbol{\mu}, \tau)\, dS\, d\tau. \tag{17}$$

In the context of the optimization problem in (1), $\mathcal{F}$ corresponds to either the objective or a constraint function.

Discretization of quantities of interest will introduce an *additional* discretization error, on top of that in the approximation of $\boldsymbol{U}$ itself. To ensure the quantity of interest discretization does not dominate, thereby lowering the global order of the scheme, it is necessary that its discretization order *matches* that of the discretization of the governing equation. Clearly, it is wasteful to discretize this to a higher order than the state equation, using a similar argument.

For these reasons, discretization of (17) will be done in a *solver-consistent* manner, i.e. the spatial and temporal discretization used for the governing equation will also be used for the quantities of interest. Define $f_h$ as the approximation of $\int_{\boldsymbol{\Gamma}} w(\boldsymbol{x},t) f(\boldsymbol{U},\boldsymbol{\mu},t)\, dS$ using the DG shape functions from the spatial discretization of the governing equations. Then, the solver-consistent spatial discretization of (17) becomes

$$\mathcal{F}_h(\boldsymbol{u},\boldsymbol{\mu},t) = \int_0^t f_h(\boldsymbol{u},\boldsymbol{\mu},\tau)\, d\tau, \tag{18}$$

ensuring the spatial integration error in the quantity of interest exactly matches that of the governing equations. Solver-consistent temporal discretization requires the semi-discrete functional in (18) be converted to an ODE, which is accomplished via differentiation of (18) with respect to $t$

$$\dot{\mathcal{F}}_h(\boldsymbol{u},\boldsymbol{\mu},t) = f_h(\boldsymbol{u},\boldsymbol{\mu},t). \tag{19}$$

Augmenting the semi-discrete governing equations with this ODE (19) yields the system of ODEs

$$\begin{bmatrix} \mathbb{M} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\mathcal{F}}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}(\boldsymbol{u},\boldsymbol{\mu},t) \\ f_h(\boldsymbol{u},\boldsymbol{\mu},t) \end{bmatrix}. \tag{20}$$

Application of the DIRK temporal discretization introduced in Section 2.3 yields the fully discrete governing equations and corresponding solver-consistent discretization of the quantity of interest (17)

$$\begin{aligned} \boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\ \mathcal{F}_h^{(n)} &= \mathcal{F}_h^{(n-1)} + \sum_{i=1}^{s} b_i f_h\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right) \\ \mathbb{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right), \end{aligned} \tag{21}$$

for $n = 1, \ldots, N_t$, $i = 1, \ldots, s$, and $\boldsymbol{u}_i^{(n)}$ is defined in (16). Finally, the functional in (17) is evaluated at time $t = T$ to yield the solver-consistent approximation of $\mathcal{F}(\boldsymbol{u},\boldsymbol{\mu},T)$

$$F(\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \ldots, \boldsymbol{k}_s^{(N_t)}) = \mathcal{F}_h^{(N_t)} \approx \mathcal{F}(\boldsymbol{u},\boldsymbol{\mu},T). \tag{22}$$

**Remarks.** In addition to the standard choice of $w(\boldsymbol{x},t) = 1$, other common choices include (a) instantaneous, spatially integrated quantities, i.e. $w(\boldsymbol{x},t) = \delta(t - t^*)$, (b) pointwise, temporally integrated quantities, i.e. $w(\boldsymbol{x},t) = \delta(\|\boldsymbol{x} - \boldsymbol{x}^*\|)$, and (c) pointwise, instantaneous quantities, i.e. $w(\boldsymbol{x},t) = \delta(\|\boldsymbol{x} - \boldsymbol{x}^*\|)\delta(t - t^*)$. where $\boldsymbol{x}^* \in \boldsymbol{\Gamma}$ and $t^* \in (0,T)$. For these specific choices of $w(\boldsymbol{x},t)$, simplifications to the above discretization can be leveraged since the spatial and/or temporal integrals reduce to an evaluation of $f$ at $\boldsymbol{x} = \boldsymbol{x}^*$ and/or $t = t^*$.

While the spatially solver-consistent discretization of quantities of interest is widely used, particularly in the context of finite element methods, temporal discretization is commonly done via low-order quadrature rules, usually the trapezoidal rule [22,4,13,10,40]. The main advantage of this solver-consistent discretization is the asymptotic discretization order of the governing equation and quantity of interest are guaranteed to exactly match, which ensures there is no wasted error in "over-integrating" one of the terms. Furthermore, solver-consistency extends the notion of adjoint-consistency that requires the discrete adjoint equations represent a consistent discretization of the continuous adjoint equations, but does not necessarily address the practical issue of evaluating the integrals that arise in each term of the finite-dimensional primal and adjoint residual. Solver-consistency ensures the integrals arising in the primal residual and quantity of interest are approximated identically, which in turn implies that each term in the adjoint equations uses the same integral approximation. The solver-consistent discretization also has the advantage of a natural and convenient implementation given the spatial and temporal discretization implementation. Finally, this method has the additional convenience of keeping a high-order accurate "current" value of the integral, i.e. at time step $n$, $\mathcal{F}_h^{(n)} \approx \int_0^{t_n} f_h(\tau)\, d\tau$ to high-order accuracy. This property does not hold for high-order numerical quadrature since $\int_0^{t_n} f_h(\tau)\,d\tau$ will involve $\boldsymbol{u}^{(n+j)}$, where $j \geq 1$ depends on the quadrature rule used.

## 3. Fully discrete, time-dependent adjoint equations

The purpose of this section is to derive an expression for the total derivative of the discrete quantity of interest $F$ in (22), which can be expanded as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}, \tag{23}$$

*that does not depend on the sensitivities of the state variables,* $\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$. Each of the $N_{\boldsymbol{\mu}}$ state variable sensitivities is the solution of a linear evolution equation of the same dimension and number of steps as the primal equation (15), rendering these quantities intractable to compute when $N_{\boldsymbol{\mu}}$ is large. Elimination of the state variable sensitivities from (23) is accomplished through introduction of the adjoint equations corresponding to the functional $F$, and the corresponding dual variables. From the derivation of the adjoint equation in Section 3.1, an expression for the reconstruction of the gradient of $F$, independent of the state variables sensitivities, follows naturally. At this point, it is emphasized that $F$ represents *any* quantity of interest whose gradient is desired, such as the optimization objective function or a constraint. This section concludes with a discussion of the advantages of the fully discrete framework in the setting of the high-order numerical scheme.

Before proceeding to the derivation of the adjoint method, the following definitions are introduced for the Runge–Kutta stage equations and state updates

$$\begin{aligned}
\tilde{\boldsymbol{r}}^{(0)}(\boldsymbol{u}^{(0)},\ \boldsymbol{\mu}) &= \boldsymbol{u}^{(0)} - \boldsymbol{u}_0(\boldsymbol{\mu}) = 0 \\
\tilde{\boldsymbol{r}}^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{u}^{(n)},\ \boldsymbol{k}_1^{(n)},\dots,\boldsymbol{k}_s^{(n)},\ \boldsymbol{\mu}) &= \boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)} - \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(i)} = 0 \\
\boldsymbol{R}_i^{(n)}(\boldsymbol{u}^{(n-1)},\boldsymbol{k}_1^{(n)},\dots,\boldsymbol{k}_i^{(n)},\boldsymbol{\mu}) &= \mathbb{M}\boldsymbol{k}_i^{(n)} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right) = 0
\end{aligned} \tag{24}$$

for $n = 1,\dots,N_t$ and $i = 1,\dots,s$. Differentiation of these expressions with respect to $\boldsymbol{\mu}$ gives rise to the *fully discrete sensitivity equations*

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(0)}} \frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}} &= 0 \\
\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}} &= 0 \\
\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^{i} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_j^{(n)}} \frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{\mu}} &= 0
\end{aligned} \tag{25}$$

where $n = 1,\dots,N_t$, $i = 1,\dots,s$, and arguments have been dropped.

### *3.1. Derivation*

The derivation of the fully discrete adjoint equations corresponding to the quantity of interest, $F$, begins with the introduction of test variables

$$\boldsymbol{\lambda}^{(0)},\ \boldsymbol{\lambda}^{(n)},\ \boldsymbol{\kappa}_i^{(n)} \in \mathbb{R}^{N_{\boldsymbol{u}}} \tag{26}$$

for $n = 1,\dots,N_t$ and $i = 1,\dots,s$. To eliminate the state sensitivities from the expression for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (23), multiply the sensitivity equations (25) by the test variables, integrate (i.e. sum in the discrete setting) over the time domain, and subtract from the expression for the gradient in (23) to obtain

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t}\sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{(0)\,T}\left[\frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(0)}}{\partial \boldsymbol{u}^{(0)}} \frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}}\right] \\ &- \sum_{n=1}^{N_t} \boldsymbol{\lambda}^{(n)\,T}\left[\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{p=1}^{s} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}}\right] \\ &- \sum_{n=1}^{N_t}\sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)\,T}\left[\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} + \sum_{j=1}^{i} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_j^{(n)}} \frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{\mu}}\right].\end{aligned} \tag{27}$$

The right side of the equality in (27) is an equivalent expression for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ *for any value of the test variables* since the terms in the brackets are zero, i.e. the sensitivity equations. Re-arrangement of terms in (27) leads to the following expression for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, where the state variable sensitivities have been isolated

$$\begin{aligned}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} &+ \left[\frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}} - \boldsymbol{\lambda}^{(N_t)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}\right] \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \boldsymbol{\lambda}^{(n)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{p=1}^{s} \boldsymbol{\kappa}_p^{(n)\,T} \frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t}\left[\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n-1)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}} - \boldsymbol{\lambda}^{(n)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}} - \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)\,T} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}\right] \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{\mu}} \\ &+ \sum_{n=1}^{N_t}\sum_{p=1}^{s}\left[\frac{\partial F}{\partial \boldsymbol{k}_p^{(n)}} - \boldsymbol{\lambda}^{(n)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_p^{(n)}} - \sum_{i=p}^{s} \boldsymbol{\kappa}_i^{(n)\,T} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{k}_p^{(n)}}\right] \frac{\partial \boldsymbol{k}_p^{(n)}}{\partial \boldsymbol{\mu}}.\end{aligned} \tag{28}$$

The dual variables, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$, which have remained arbitrary to this point, are chosen as the solution to the following equations

$$\begin{aligned}\frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}^T \boldsymbol{\lambda}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\ \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\lambda}^{(n)} + \frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\lambda}^{(n-1)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T - \sum_{i=1}^{s} \frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}^T \boldsymbol{\kappa}_i^{(n)} \\ \sum_{j=i}^{s} \frac{\partial \boldsymbol{R}_j^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T \boldsymbol{\kappa}_j^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T - \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}^T \boldsymbol{\lambda}^{(n)}\end{aligned} \tag{29}$$

for $n = 1, \dots, N_t$ and $i = 1, \dots, s$. These are the *fully discrete adjoint equations* corresponding to the primal evolution equations in (24) and quantity of interest $F$. Defining the dual variables as the solution of the adjoint equations in (29), the expression for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (28) reduces to

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \boldsymbol{\lambda}^{(n)\,T} \frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t}\sum_{p=1}^{s} \boldsymbol{\kappa}_p^{(n)\,T} \frac{\partial \boldsymbol{R}_p^{(n)}}{\partial \boldsymbol{\mu}}, \tag{30}$$

which is *independent* of the state sensitivities. Finally, elimination of the auxiliary terms, $\tilde{\boldsymbol{r}}^{(n)}$ and $\boldsymbol{R}_i^{(n)}$, in equations (28) and (29) through differentiation of their expressions in (24) gives rise to the adjoint equations

$$\begin{aligned}\boldsymbol{\lambda}^{(N_t)} &= \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T \\ \boldsymbol{\lambda}^{(n-1)} &= \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_i^{(n)} \\ \mathbb{M}^T \boldsymbol{\kappa}_i^{(n)} &= \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_j^{(n)}\end{aligned} \tag{31}$$

for $n = 1, \dots, N_t$ and $i = 1, \dots, s$ and the expression for gradient reconstruction, independent of state sensitivities,

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n), \tag{32}$$

specialized to the case of a DIRK temporal discretization. From inspection of (32), it is clear that the initial condition sensitivity $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ is the only sensitivity term required to reconstruct $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$. The presence of this term does not destroy the efficiency of the adjoint method for two reasons: (a) only matrix–vector products with $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}^T$ are required and (b) the parametrization of the initial condition is either known analytically (uniform flow, zero freestream, independent of $\boldsymbol{\mu}$, etc.) or is the solution of some nonlinear system of equations (most likely the steady-state equations). In the first case, $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ can be computed analytically once $\boldsymbol{\lambda}^{(0)}$ is known. The next section details efficient computation of $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ using the adjoint method of the steady-state problem.

### 3.2. Parametrization of initial condition

Suppose the initial condition $\boldsymbol{u}_0(\boldsymbol{\mu})$ is defined as the solution of the nonlinear system of equations – whose Jacobian is invertible at $\boldsymbol{u}_0(\boldsymbol{\mu})$ – which is most likely the fully discrete steady-state form of the governing equations

$$\boldsymbol{R}(\boldsymbol{u}_0(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0. \tag{33}$$

Differentiating with respect to the parameter $\boldsymbol{\mu}$ leads to the expansion

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} = 0, \tag{34}$$

where arguments have been dropped for brevity. Assuming the Jacobian matrix is invertible, multiply the preceding equation by the $\boldsymbol{\lambda}^{(0)}$ and rearrange to obtain

$$-\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} = \left[ \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}^{-T} \boldsymbol{\lambda}^{(0)} \right]^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}. \tag{35}$$

This reveals the term $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ can be computed at the cost of one linear system solve of the form $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}_0}^T \boldsymbol{v} = \boldsymbol{\lambda}^{(0)}$ and an inner product $\boldsymbol{v}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}$. The only operation whose cost scales with the size of $\boldsymbol{\mu}$ is the evaluation of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\mu}}$ and subsequent inner product. Given this exposition on the fully discrete, time-dependent adjoint method and the discrete adjoint method for computing $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$, a discussion is provided detailing the advantages of the fully discrete framework when computing gradients of output quantities before discussing implementation details in Section 4.

### 3.3. Benefits of fully discrete framework

In the context of optimization, the fully discrete adjoint method is advantageous compared to the continuous or semi-discrete version as it is guaranteed that the resulting derivatives will be consistent with the quantity of interest, $F$. This emanates from the fact that in the fully discrete setting, the *discretization errors are also differentiated*. This property is practically relevant as convergence guarantees and convergence rates of many black-box optimizers are heavily dependent on consistent gradients of optimization functionals.

Additionally, when Runge–Kutta schemes are chosen for the temporal discretization, the fully discrete framework is particularly advantageous since the *stages* are rarely invariant with respect to the direction of time, that is to say,

$$\nexists i, j \in \{1, \dots, s\} \quad \text{such that} \quad t_{n-1} + c_i \Delta t_n = t_n - c_j \Delta t_n, \tag{36}$$

where $c$ is from the Butcher tableau. Temporal invariance of a Runge–Kutta scheme, as defined in (36) is significant when computing adjoint variables. During the primal solve, $\boldsymbol{u}$ will be computed at $t_n$ for $n = 1, \dots, N$ and its stage values at $t_{n-1} + c_i \Delta t_n$ for $n = 1, \dots, N$ and $i = 1, \dots, s$. If the same RK scheme is applied to integrate the *semi-discrete* adjoint equations backward in time, the primal solution will be required at $t_n - c_i \Delta t_n$ for $n = 1, \dots, N$ and $i = 1, \dots, s$. Due to condition (36), the solution to the primal problem was not computed during the forward solve. Obtaining the primal solution at this time requires interpolation, which complicates the implementation, degrades the accuracy of the computed adjoint variables, and destroys discrete consistency of the computed gradients. This issue does not arise in the fully discrete setting as only terms computed during the primal solve appear in the adjoint equations, by construction.

The next section is devoted to detailing an efficient and modular implementation of the fully discrete adjoint method on deforming domains.

**Algorithm 1** Primal solution: functional evaluation.

```
Input: Initial condition, $\boldsymbol{u}^{(0)}$; parameter configuration, $\boldsymbol{\mu}$
Output: Integrated output quantity, $F = \mathcal{F}_h^{(N_t)}$, and primal state quantities, $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$ for $n = 1, \dots, N_t$ and $i = 1, \dots, s$
Initialize: $\mathcal{F}_h^{(0)} = 0$
for $n = 1, \dots, N_t$ do
    for $i = 1, \dots, s$ do
        Solve (15) for $\boldsymbol{k}_i^{(n)}$
            $$\mathbb{M}\boldsymbol{k}_i^{(n)} = \Delta t \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t\right)$$
        where $\boldsymbol{u}_i^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij}\boldsymbol{k}_j^{(n)}$
        Write $\boldsymbol{k}_i^{(n)}$ to disk
    end for
    Update $\boldsymbol{u}$ according to (15)
        $$\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)}$$
    Update $\mathcal{F}_h$ according to (21)
        $$\mathcal{F}_h^{(n)} = \mathcal{F}_h^{(n-1)} + \sum_{i=1}^{s} b_i f\left(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_i \Delta t_n\right)$$
    Write $\boldsymbol{u}^{(n)}$ to disk
end for
```

## 4. Implementation

Implementation of the fully discrete adjoint method introduced in Section 3 relies on the computation of the following terms from the spatial discretization

$$\mathbb{M}, \boldsymbol{r}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T, \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}, f_h, \frac{\partial f_h}{\partial \boldsymbol{u}}, \frac{\partial f_h}{\partial \boldsymbol{\mu}}. \tag{37}$$

Here, $\mathbb{M}$ is the mass matrix of the semi-discrete conservation law, and $\boldsymbol{r}$ is the spatial residual vector with derivatives $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}$ (Jacobian) and $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}$. As in the previous section, $f_h$ is the discretization of the spatial integral of the output quantity of interest with derivatives $\frac{\partial f_h}{\partial \boldsymbol{u}}$ and $\frac{\partial f_h}{\partial \boldsymbol{\mu}}$. The mass matrix, spatial flux, Jacobian of spatial flux, and output quantity are standard terms required by an implicit solver and will not be considered further. The Jacobian transpose is explicitly mentioned as additional implementational effort is required when performing parallel matrix transposition. The derivatives with respect to $\boldsymbol{\mu}$ are rarely required outside adjoint method computations and will be considered further in Section 4.1. As indicated in Section 2.2, all relevant derivatives of the mass matrix are zero since it is independent of time, parameter, and state variable, which is an artifact of the transformation to a fixed reference domain.

The parallel implementation of all semi-discrete quantities in (37) is performed using domain decomposition, where each processor contains a subset of the elements in the mesh, including a halo of elements to be communicated with neighbors [41]. Linear systems of the form

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\mathbf{x} = \mathbf{b} \qquad \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T \mathbf{x} = \mathbf{b}$$

are solved in parallel using a GMRES solver with a block Incomplete-LU (ILU) preconditioner.

Given the availability of all terms in (37), the solution of the primal problem and integration of the output quantity $F$ is given in Algorithm 1. The solution of the corresponding fully discrete adjoint equation, and reconstruction of the gradient of $F$, is given in Algorithm 2.

A well-documented implementational issue corresponding to the unsteady adjoint method pertains to storage and I/O demands. The adjoint equations are solved backward in time and require the solution of the primal problem at each of the corresponding steps/stages. Therefore, the adjoint computations cannot begin until all primal states have been computed. Additionally, this implies all primal states must be stored since they will be required in *reverse* order during the adjoint computation. For most problems, storing all primal states in memory will be infeasible, requiring disk I/O, which must be performed in parallel to ensure parallel scaling is maintained. There have been a number of strategies to minimize the required I/O operations, such as local-in-time adjoint strategies [42] and checkpointing [43–45]. For the DG-ALE method in this work, the cost of I/O was not significant compared to the cost of assembly and solving the linearized system of equations.

**Algorithm 2** Dual solution: gradient evaluation.

```
Input: Primal state quantities, $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$ for $n = 1, \dots, N_t$ and $i = 1, \dots, s$; initial condition sensitivity, $\frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{\mu}}$; parameter configuration, $\boldsymbol{\mu}$
Output: Gradient of integrated output quantity, $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, and dual state quantities, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$ for $n = 1, \dots, N_t$ and $i = 1, \dots, s$
Read primal solution $\boldsymbol{u}^{(N_t)}$ from disk
$\boldsymbol{\lambda}^{(N_t)} = \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T$
Initial gradient of $F$ with partial derivative and initial condition sensitivity
    $$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}^{(0)T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$$
for $n = N_t, \dots, 1$ do
    Read primal solution $\boldsymbol{u}^{(n-1)}$ from disk
    for $i = s, \dots, 1$ do
        Read primal solution $\boldsymbol{k}_i^{(n)}$ from disk
        Solve (31) for $\boldsymbol{\kappa}_i^{(n)}$
            $$\mathbb{M}^T \boldsymbol{\kappa}_i^{(n)} = \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}}^T + b_i \boldsymbol{\lambda}^{(n)} + \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_j^{(n)}, \boldsymbol{\mu}, t_{n-1} + c_j \Delta t_n)^T \boldsymbol{\kappa}_j^{(n)}$$
        Update $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ according to (32)
            $$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} + \Delta t_n \boldsymbol{\kappa}_i^{(n)T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n)$$
    end for
    Update $\boldsymbol{\lambda}$ according to (31)
        $$\boldsymbol{\lambda}^{(n-1)} = \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}_i^{(n)}, \boldsymbol{\mu}, t_i + c_i \Delta t_n)^T \boldsymbol{\kappa}_i^{(n)}$$
end for
```

In this work, the 3DG software [46] was used for the high-order DG-ALE scheme. The temporal discretization and unsteady adjoint method were implemented in the Model Order Reduction Testbed (MORTestbed) [47,48] code-base, which was used to wrap 3DG such that all data structures, and thus all parallel capabilities, were inherited.

### 4.1. *Partial derivatives of residuals and output quantities*

This section details computation of partial derivatives of the residual, $\boldsymbol{r}$, and the output quantity, $f_h$, with respect to the parameter $\boldsymbol{\mu}$. The DG-ALE discretizations of Section 2.2, with and without GCL augmentation, are considered separately as the implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$ requires special treatment.

#### 4.1.1. *Without GCL augmentation*

When the GCL augmentation is not considered, the dependence of $\boldsymbol{r}$ and $f_h$ on the parameter $\boldsymbol{\mu}$ is solely due to the domain parametrization. Therefore, the following expansion of the partial derivatives with respect to $\boldsymbol{\mu}$ is exploited

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}} \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} \qquad \frac{\partial f_h}{\partial \boldsymbol{\mu}} = \frac{\partial f_h}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f_h}{\partial \dot{\boldsymbol{x}}} \frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} \tag{38}$$

where $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ are determined solely from the domain parametrization in Section 4.2 and the terms

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}, \frac{\partial f_h}{\partial \boldsymbol{x}}, \frac{\partial f_h}{\partial \dot{\boldsymbol{x}}} \tag{39}$$

are determined from the form of the governing equations and spatial discretization outlined in Section 2. From the expressions in (38), the terms in (39) are not explicitly required in matrix form, rather matrix–vector products with $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ from Section 4.2 are required.

#### 4.1.2. *With GCL augmentation*

For the DG-ALE scheme with GCL augmentation, the dependence of $\boldsymbol{r}$ and $f$ on the parameter $\boldsymbol{\mu}$ arises from two sources, the domain parametrization and the implicit dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$. Therefore, the chain rule expansions in (38) must include an additional term to account for the dependence of $\bar{\mathbf{g}}$ on $\boldsymbol{\mu}$

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}} \qquad \frac{\partial f}{\partial \boldsymbol{\mu}} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \dot{\boldsymbol{x}}}\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}} + \frac{\partial f}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}. \tag{40}$$

Similar to the previous section, the terms $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ are determined solely from the domain parametrization in Section 4.2 and

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{x}}}, \frac{\partial \boldsymbol{r}}{\partial \bar{\mathbf{g}}}, \frac{\partial f}{\partial \boldsymbol{x}}, \frac{\partial f}{\partial \dot{\boldsymbol{x}}}, \frac{\partial f}{\partial \bar{\mathbf{g}}} \tag{41}$$

are determined from the form of the governing equations and spatial discretization in Section 2. The only remaining term $\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}$ is defined as the solution of the following ODE

$$\mathbb{M}_{\bar{\mathbf{g}}}\frac{\partial}{\partial t}\left(\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}}\right) = \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \bar{\mathbf{g}}}\frac{\partial \bar{\mathbf{g}}}{\partial \boldsymbol{\mu}} = \frac{\partial \boldsymbol{r}_{\bar{\mathbf{g}}}}{\partial \boldsymbol{\mu}}, \tag{42}$$

obtained by direct differentiation of (12). The last equality uses the fact that $\boldsymbol{r}_{\bar{\mathbf{g}}}$ is independent of $\bar{\mathbf{g}}$, which can be deduced from examination of the governing equation for $\bar{g}$ (6). Equation (42) is discretized with the same DIRK scheme used for the temporal discretization of the state equation.

**Remarks.** The special treatment of $\bar{\mathbf{g}}$ detailed in this section, including integration of the sensitivity equations (42), can be avoided by considering the ODEs in (12) directly without leveraging the fact that the $\bar{\mathbf{g}}$ equation is independent of $\boldsymbol{u}_{\bar{X}}$. This implies the state vector will contain an additional unknown for $\bar{g}$ for each DG node. This increases the cost of a primal and dual solve, but simplifies the adjoint derivation and implementation, allowing the form in Section 4.1.1 to be used directly.

### *4.2. Time-dependent, parametrized domain deformation*

A crucial component of the fully discrete adjoint method on deforming domains is a time-dependent parametrization of the domain, amenable to parallel implementation. A parallel implementation is required as domain deformation will involve operations on the entire computational mesh and will be queried at every stage of each time step of both the primal and dual solves, according to Algorithms 1 and 2. In this work, the domain parametrization is required to be sufficiently general to handle shape deformation, as well as kinematic motion. Additionally, the domain deformation must be sufficiently smooth to ensure sufficient regularity of the transformed solution, and the spatial and temporal derivatives must be analytically available for fast, accurate computation of the deformation gradient, $\boldsymbol{G}$, and velocity, $\boldsymbol{v}_{\boldsymbol{X}}$, of the mapping, $\mathcal{G}$.

The domain deformation will be defined by the superposition of a rigid body motion and a spatially varying deformation. To avoid large mesh velocities at the far-field, which could arise from rigid rotations of the body, the blending maps of [17] are used. First, define a spatial configuration consisting of a rigid body motion ($\boldsymbol{Q}\,(\boldsymbol{\mu}, t)$, $\boldsymbol{v}(\boldsymbol{\mu}, t)$) and deformation ($\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t)$) to the *reference* domain

$$\boldsymbol{X}' = \boldsymbol{Q}\,(\boldsymbol{\mu}, t)\boldsymbol{X} + \boldsymbol{v}(\boldsymbol{\mu}, t) + \boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t), \tag{43}$$

which completely defines the *physical* motion of the body. This physical configuration is blended with the reference configuration according to

$$\boldsymbol{x} = (1 - b(d(\boldsymbol{X})))\boldsymbol{X}' + b(d(\boldsymbol{X}))\boldsymbol{X} \tag{44}$$

where $d(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{X}_0\| - R_0$ is the signed distance from the origin $\boldsymbol{X}_0$ to the circle of radius $R_0$ centered at $\boldsymbol{X}_0$ and

$$b(s) = \begin{cases} 0, & \text{if } s < 0 \\ 1, & \text{if } s > R_1 \\ r(s/R_1), & \text{otherwise} \end{cases} \tag{45}$$

where $r(s) = 3s^2 - 2x^3$ for a cubic blending and $r(s) = 10s^3 - 15s^4 + 6s^5$ for a quintic blending. Spatial blending of this form ensures the desired physical motion of the body, $\boldsymbol{X}'$ is exactly achieved within a radius $R_0$ of the origin. Further, there is no deformation outside a radius $R_0 + R_1$ of the origin. In the annulus about the origin with inner radius $R_0$ and outer radius $R_0 + R_1$, the spatial configuration is blended smoothly between these two spatial configurations.

The specific form of $\boldsymbol{Q}\,(\boldsymbol{\mu}, t)$, $\boldsymbol{v}(\boldsymbol{\mu}, t)$, and $\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t)$ is problem-specific and will be deferred to Section 5. Assuming these terms are known analytically, the specific form of $\boldsymbol{G} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$, $\boldsymbol{v}_{\boldsymbol{X}} = \dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial t}$, $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\mu}}$, and $\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\mu}}$ can be easily computed.

**Table 2**
Butcher Tableau for 3-stage, 3rd order DIRK scheme [24] $\alpha = 0.435866521508459$, $\gamma = -\frac{6\alpha^2-16\alpha+1}{4}$, $\omega = \frac{6\alpha^2-20\alpha+5}{4}$.

| | | | |
|---|---|---|---|
| $\alpha$ | $\alpha$ | | |
| $\frac{1+\alpha}{2}$ | $\frac{1+\alpha}{2}-\alpha$ | $\alpha$ | |
| $1$ | $\gamma$ | $\omega$ | $\alpha$ |
| | $\gamma$ | $\omega$ | $\alpha$ |

## 5. Applications

In this section, the high-order numerical discretization of a system of conservation laws and corresponding adjoint method is applied to the isentropic compressible Navier–Stokes equations to solve optimal control and shape optimization problems using gradient-based optimization techniques. The compressible Navier–Stokes equations take the form

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0, \tag{46}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) = +\frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3, \tag{47}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}\big(u_j(\rho E + p)\big) = -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(u_j \tau_{ij}), \tag{48}$$

in $v(\boldsymbol{\mu}, t)$ where $\rho$ is the fluid density, $u_1, u_2, u_3$ are the velocity components, and $E$ is the total energy. The viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \qquad \text{and} \qquad q_j = -\frac{\mu}{\mathrm{Pr}}\frac{\partial}{\partial x_j}\left(E + \frac{p}{\rho} - \frac{1}{2}u_k u_k\right). \tag{49}$$

Here, $\mu$ is the viscosity coefficient and $\mathrm{Pr} = 0.72$ is the Prandtl number which we assume to be constant. For an ideal gas, the pressure $p$ has the form

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2}u_k u_k\right), \tag{50}$$

where $\gamma$ is the adiabatic gas constant. Furthermore, the entropy of the system is assumed constant, which is tantamount to the flow being adiabatic and reversible. For a perfect gas, the entropy is defined as

$$s = p/\rho^{\gamma}. \tag{51}$$

Using (51) to explicitly relate the pressure and density, the energy equation becomes redundant. This effectively reduces the square system of PDEs of size $n_{sd}+2$ to one of size $n_{sd}+1$, where $n_{sd}$ is the number of spatial dimensions. It can be shown, under suitable assumptions, that the solution of the isentropic approximation of the Navier–Stokes equations converges to the solution of the incompressible Navier–Stokes equations as the Mach number goes to 0 [49–51].

The DG-ALE scheme introduced in Section 2 is used for the spatial discretization of the system of conservation laws with polynomial order $p = 3$ and a diagonally implicit Runge–Kutta scheme for the temporal discretization. The DG-ALE scheme uses the Roe flux [52] for the inviscid numerical flux and the Compact DG flux [46] for the viscous numerical flux. The Butcher tableau for the three-stage, third-order DIRK scheme considered in this work is given in Table 2.

The instantaneous quantities of interest for a body, defined by the surface $\boldsymbol{\Gamma}$, take the following form

$$\begin{aligned}
\mathcal{F}_x(\boldsymbol{U}, \boldsymbol{\mu}, t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \boldsymbol{e}_1 \, dS & \mathcal{F}_y(\boldsymbol{U}, \boldsymbol{\mu}, t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \boldsymbol{e}_2 \, dS \\
\mathcal{P}(\boldsymbol{U}, \boldsymbol{\mu}, t) &= \int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \dot{\boldsymbol{x}} \, dS & \mathcal{P}_x(\boldsymbol{U}, \boldsymbol{\mu}, t) &= \int_{\boldsymbol{\Gamma}} \dot{x} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \boldsymbol{e}_1 \, dS \\
\mathcal{P}_y(\boldsymbol{U}, \boldsymbol{\mu}, t) &= \int_{\boldsymbol{\Gamma}} \dot{y} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \cdot \boldsymbol{e}_2 \, dS & \mathcal{P}_\theta(\boldsymbol{U}, \boldsymbol{\mu}, t) &= -\int_{\boldsymbol{\Gamma}} \dot{\theta} \boldsymbol{f}(\boldsymbol{U}, \boldsymbol{\mu}, t) \times (\boldsymbol{x} - \boldsymbol{x}_0) \, dS
\end{aligned} \tag{52}$$

where $\boldsymbol{f} \in \mathbb{R}^{n_{sd}}$ is the force imparted by the fluid on the body, $\boldsymbol{e}_i$ is the $i$th canonical basis vector in $\mathbb{R}^{n_{sd}}$, $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are the position and velocity of a point on the surface $\boldsymbol{\Gamma}$, and $x$, $y$, $\theta$, $\dot{x}$, $\dot{y}$, $\dot{\theta}$ define the motion of the reference point, $\boldsymbol{x}_0$ (the 1/3-chord of the airfoil, in this case); see Fig. 2. The $\mathcal{F}_x$ and $\mathcal{F}_y$ terms correspond to the total $x$- and $y$-directed forces on the body and $\mathcal{P}$ is the total power exerted on the body by the fluid. The total power $\mathcal{P}$ is broken into its translational, $\mathcal{P}_x$ and $\mathcal{P}_y$, and rotational, $\mathcal{P}_\theta$, components. For a 2D rigid body motion, an additive relationship among these terms holds

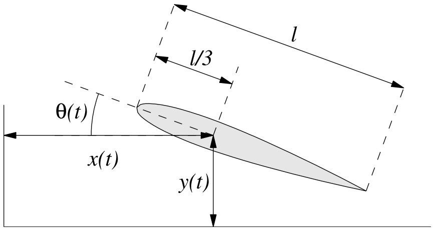

**Fig. 2.** Airfoil

**Table 3**
Summary of parametrizations considered in Section 5.1. The number of clamped cubic spline knots used to discretize $x(t)$, $y(t)$, and $\theta(t)$ are $m_x+1$, $m_y+1$, and $m_\theta+1$, respectively. PI freezes the rigid body translation ($m_x=m_y=0$) and optimizes over only the rotation ($m_\theta \neq 0$). PII optimizes over all rigid body degrees of freedom ($m_x=m_y=m_\theta \neq 0$).

| ID | PI | | | | PII | | | |
|---|---|---|---|---|---|---|---|---|
| | $m_x$ | $m_y$ | $m_\theta$ | $N_{\boldsymbol{\mu}}$ | $m_x$ | $m_y$ | $m_\theta$ | $N_{\boldsymbol{\mu}}$ |
| 1 | 0 | 0 | 2 | 3 | 2 | 2 | 2 | 9 |
| 2 | 0 | 0 | 6 | 7 | 6 | 6 | 6 | 21 |
| 3 | 0 | 0 | 10 | 11 | 10 | 10 | 10 | 33 |
| 4 | 0 | 0 | 15 | 16 | 15 | 15 | 15 | 48 |
| 5 | 0 | 0 | 25 | 26 | 25 | 25 | 25 | 78 |
| 6 | 0 | 0 | 50 | 51 | 50 | 50 | 50 | 153 |
| 7 | 0 | 0 | 100 | 101 | 100 | 100 | 100 | 303 |

$$\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t)=\mathcal{P}_x(\boldsymbol{U},\boldsymbol{\mu},t)+\mathcal{P}_y(\boldsymbol{U},\boldsymbol{\mu},t)+\mathcal{P}_\theta(\boldsymbol{U},\boldsymbol{\mu},t). \tag{53}$$

The negative sign is included in the definition of $\mathcal{P}_\theta$ due to the clockwise definition of $\theta$ in Fig. 2. In the remainder of this document, a superscript $h$ will be used to denote the high-order DG approximation to these spatial integrals that constitute the instantaneous quantities of interest, e.g., $\mathcal{P}^h(\boldsymbol{u},\boldsymbol{\mu},t)$ is the high-order approximation of $\mathcal{P}(\boldsymbol{U},\boldsymbol{\mu},t)$, where $\boldsymbol{u}$ is the semi-discrete approximation of $\boldsymbol{U}$. Temporal integration of the instantaneous quantities of interest leads to the integrated quantities of interest

$$\begin{aligned}
\mathcal{J}_x(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1\,dS\,dt & \mathcal{J}_y(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2\,dS\,dt \\
\mathcal{W}(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\dot{\boldsymbol{x}}\,dS\,dt & \mathcal{W}_x(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \dot{x}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_1\,dS\,dt \\
\mathcal{W}_y(\boldsymbol{U},\boldsymbol{\mu}) &= \int_0^T\int_{\boldsymbol{\Gamma}} \dot{y}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\cdot\boldsymbol{e}_2\,dS\,dt & \mathcal{W}_\theta(\boldsymbol{U},\boldsymbol{\mu}) &= -\int_0^T\int_{\boldsymbol{\Gamma}} \dot{\theta}\boldsymbol{f}(\boldsymbol{U},\boldsymbol{\mu},t)\times(\boldsymbol{x}-\boldsymbol{x}_0)\,dS\,dt
\end{aligned} \tag{54}$$

which will be used as optimization functionals in subsequent sections. The terms $\mathcal{J}_x$ and $\mathcal{J}_y$ are the $x$- and $y$-directed impulse the fluid exerts on the airfoil, respectively, $\mathcal{W}$ is the total work done on the airfoil by the fluid, and $\mathcal{W}_x$, $\mathcal{W}_y$, and $\mathcal{W}_\theta$ are the translational and rotational components of the total work. The fully discrete, high-order approximation of the integrated quantities of interest (DG in space, DIRK in time) will be denoted with the corresponding Roman symbol, e.g., $W(\boldsymbol{u}^{(0)},\dots,\boldsymbol{u}^{(N_t)},\boldsymbol{k}_1^{(n)},\dots,\boldsymbol{k}_s^{(n)},\boldsymbol{\mu})$ is the fully discrete approximation of $\mathcal{W}(\boldsymbol{U},\boldsymbol{\mu})$.

### *5.1. Energetically optimal trajectory of 2D airfoil in compressible, viscous flow*

In this section, the high-order, time-dependent PDE-constrained optimization framework introduced in this document is applied to find the energetically optimal trajectory of a 2D NACA0012 airfoil with chord length $l=1$ and zero-thickness trailing edge. The governing equations are the 2D compressible, isentropic Navier–Stokes equations. The mission of the airfoil is to move a distance of $-1.5$ units horizontally and 1.5 units vertically in $T=4$ units of time, with the restriction that $\theta(0)=\theta(T)=0$, i.e. the angle of attack at the initial and final time is zero. Additionally, to ensure smoothness of the motion and avoid non-physical transients, $\dot{x}(0)=\dot{x}(T)=\dot{y}(0)=\dot{y}(T)=\dot{\theta}(0)=\dot{\theta}(T)=0$ are enforced. The goal is to determine the trajectory $x(t)$, $y(t)$, $\theta(t)$ of the airfoil that minimizes the total energy required to complete the mission, i.e.,

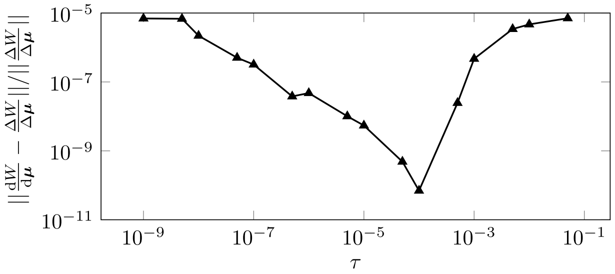

**Fig. 3.** Verification of adjoint-based gradient with fourth-order centered finite difference approximation, for a range of finite intervals, $\tau$, for the total work $W$ – the objective function in (56) – for parametrization PII (Table 3). The computed gradient match the finite difference approximation to about 10 digits of accuracy before round-off errors degrade the accuracy.

$$\begin{aligned} \underset{\boldsymbol{U},\,\boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{W}(\boldsymbol{U},\boldsymbol{\mu}) \\ \text{subject to} \quad & x(0)=\dot{x}(0)=\dot{x}(T)=0,\ x(T)=-1.5 \\ & y(0)=\dot{y}(0)=\dot{y}(T)=0,\ y(T)=1.5 \\ & \theta(0)=\theta(T)=\dot{\theta}(0)=\dot{\theta}(T)=0 \\ & \frac{\partial \boldsymbol{U}}{\partial t}+\nabla\cdot\boldsymbol{F}(\boldsymbol{U},\nabla\boldsymbol{U})=0 \quad \text{in } v(\boldsymbol{\mu},t). \end{aligned} \tag{55}$$

The trajectory of the airfoil – $x(t)$, $y(t)$, and $\theta(t)$ – is discretized via clamped cubic splines with $m_x+1$, $m_y+1$, and $m_\theta+1$ knots, respectively. The knots are uniformly spaced between 0 and $T$ in the $t$-dimension and the knot values are optimization parameters. Table 3 summarizes two parametrizations considered in this section: (PI) the translational degrees of freedom – $x(t)$ and $y(t)$ – are frozen at their nominal value in Fig. 5 and the rotational degree of freedom – $\theta(t)$ – is parametrized with a $m_\theta+1$-knot clamped cubic spline and (PII) all rigid body modes are parametrized with clamped cubic splines. The 7 IDs in Table 3 correspond to levels of refinement of the given parametrization with ID = 1 being the coarsest parametrization and ID = 7 the finest. With this parametrization of the airfoil kinematics, spatial and temporal discretization with the high-order scheme of Section 2 leads to the fully discrete version of the optimization problem in (55)

$$\begin{aligned} \underset{\substack{\boldsymbol{u}^{(0)},\,\dots,\,\boldsymbol{u}^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\,\dots,\,\boldsymbol{k}_s^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & W(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \\ \text{subject to} \quad & x(0)=\dot{x}(0)=\dot{x}(T)=0,\ x(T)=-1.5 \\ & y(0)=\dot{y}(0)=\dot{y}(T)=0,\ y(T)=1.5 \\ & \theta(0)=\theta(T)=\dot{\theta}(0)=\dot{\theta}(T)=0 \\ & \boldsymbol{u}^{(0)}=\boldsymbol{u}_0 \\ & \boldsymbol{u}^{(n)}=\boldsymbol{u}^{(n-1)}+\sum_{i=1}^{s} b_i\boldsymbol{k}_i^{(n)} \\ & \mathbb{M}\boldsymbol{k}_i^{(n)}=\Delta t_n\boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1}+c_i\Delta t_n\right). \end{aligned} \tag{56}$$

Before considering the optimization problem (56), the proposed adjoint method for computing gradients of quantities of interest on the manifold of fully discrete, high-order solutions of the conservation law (15) is verified against a fourth-order finite difference approximation. The finite difference approximation to gradients on the aforementioned manifold requires finding the solution of the fully discretized governing equations *at perturbations* about the nominal parameter configuration in Fig. 5. To mitigate round-offs errors as much as possible in the finite difference computation, the number of time steps was reduced to 10 and only half the trajectory was simulated. Fig. 3 shows the relative error between the gradients computed via the adjoint method and this finite difference approximation for a sweep of finite difference intervals, $\tau$. A relative error of nearly $10^{-11}$ is observed for a finite difference step of $\tau=10^{-4}$. As expected, the error starts to increase after $\tau$ drops too small due to the trade-off between finite difference accuracy and roundoff error.

A brief study of the convergence behavior of the underlying DG scheme is provided in Fig. 4 to justify our use of $p=3$. In this study, no attempt is made to quantify convergence rates due to the sharp trailing edge of the NACA0012 that limits regularity of the solution. From Fig. 4, the higher order methods outperform the lower order ones in terms of error – in the work, $W$, and $x$-impulse, $J_x$ – for a given number of degrees of freedom ($N_{\boldsymbol{u}}$) and CPU time. All timings were obtained using 12 CPU cores.

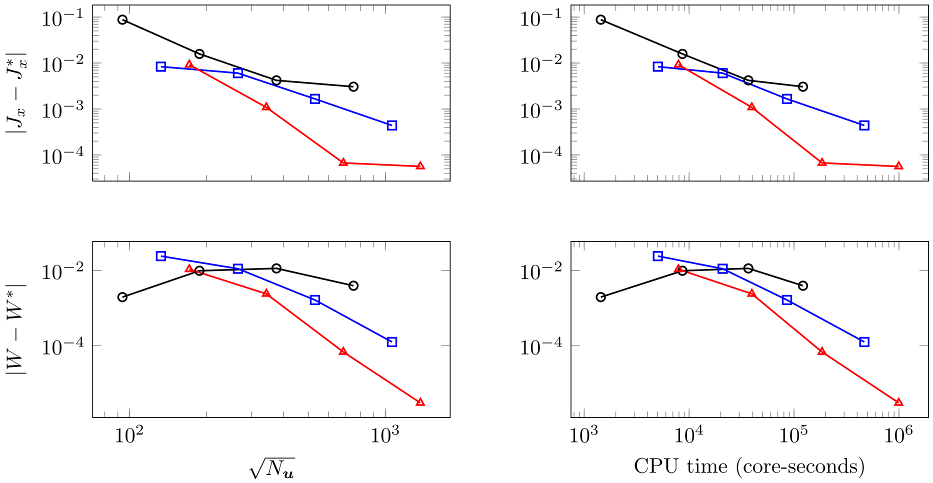

**Fig. 4.** Spatial convergence in quantities of interest for the DG-ALE-DIRK discretization for $p = 1$ (—○—), $p = 2$ (—□—), $p = 3$ (—△—). All schemes use the DIRK3 temporal discretization (Table 2) with 1000 time steps to mitigate temporal error. The reference values $W^*$ and $J_x^*$ are produced from a simulation consisting of 250368 $p = 4$ elements and 1000 DIRK3 time steps.

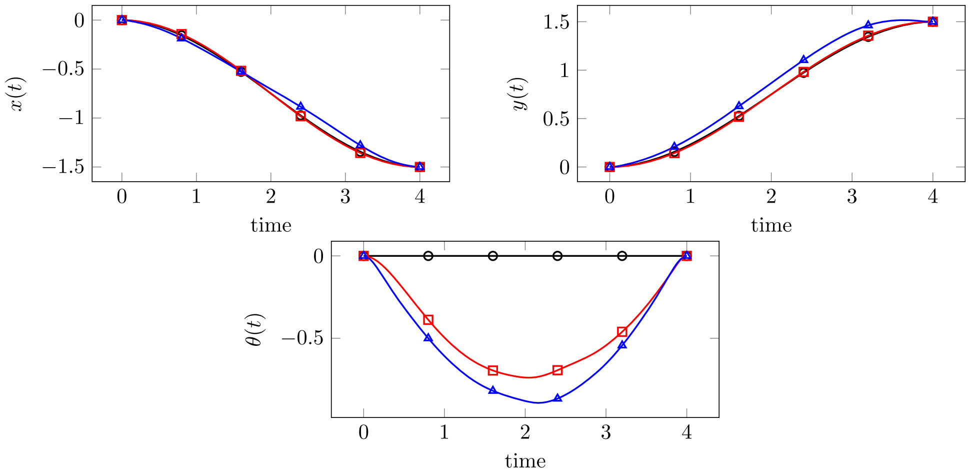

**Fig. 5.** Trajectories of $x(t)$, $y(t)$, and $\theta(t)$ at initial guess (—○—), solution of (56) under parametrization PI (—□—), and solution of (56) under parametrization PII (—△—) for ID = 7.

With this verification of the adjoint-based gradients and high-order accuracy, attention is turned to the optimization problem in (56). The optimization solver used in this section is L-BFGS-B [53], a bound-constrained, limited-memory BFGS algorithm. Fig. 5 contains the initial guess for the optimization problem in (56) as well as its solution under both parametrization, PI and PII, at the finest level of refinement (ID = 7). The initial guess for the optimization problem is a pure translational motion with $\theta(t) = 0$. The solution under parametrization PI freezes the translational motion at its nominal value and incorporates rotational motion. The solution under parametrization PII increases the amplitude of the rotation, flattens the trajectory of $x(t)$, and incorporates an overshoot in $y(t)$ before settling to the required location, as compared to the optimal solution corresponding to PI.

The instantaneous quantities of interest for the nominal trajectory and solution of (56) under parametrizations PI and PII are included in Fig. 6. It is clear that the optimal solution under both parametrizations result in a time history of the total power that is uniformly closer to 0 than that at the nominal trajectory, which is expected since $W$ is the objective function. With the exception of the edges of the time interval, the total power time history for the optimal solution under parametrization PII is uniformly closer to 0 than that of PI. The same observation holds for the power due to the translational motion, $\mathcal{P}_x^h$ and $\mathcal{P}_y^h$. Whereas the total power corresponding to the nominal trajectory is due solely to the translational motion (since there is no rotation), the optimal solutions exchange large amounts of translational power for a small amount of rotational power. These observations can also be verified in Table 4 which summarizes the optimal values of the integrated quantities of interest.

The convergence of the total work, i.e., the objective function of the optimization problem, with iterations of the optimization solver is summarized in Fig. 7 (left). Both parametrizations are included and iterations are agglomerated over all

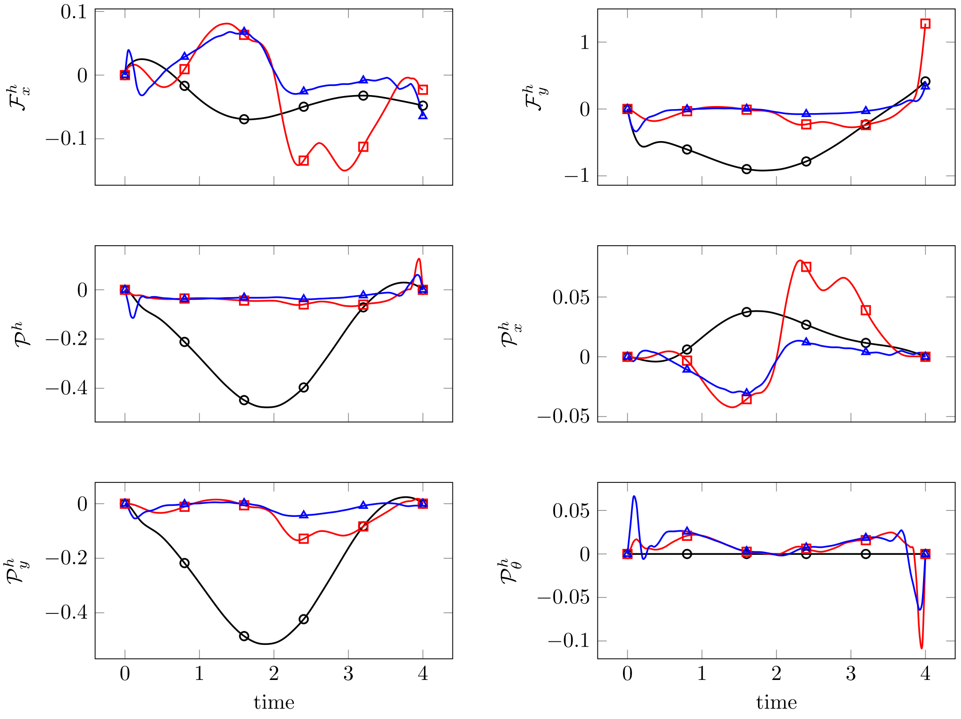

**Fig. 6.** Time history of instantaneous quantities of interest ($x$-directed force – $\mathcal{F}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, $y$-directed force – $\mathcal{F}_y^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, total power – $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, $x$-translational power – $\mathcal{P}_x^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, $y$-translational power – $\mathcal{P}_y^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, rotational power – $\mathcal{P}_\theta^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$) at initial guess (—○—), solution of (56) under parametrization PI (—□—), and solution of (56) under parametrization PII (—▲—) for ID = 7.

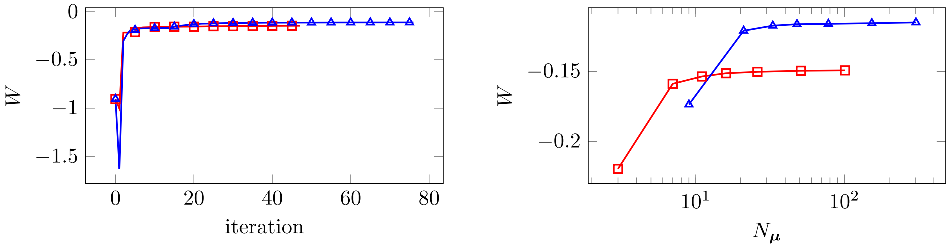

**Fig. 7.** *Left*: Convergence of total work $W$ with optimization iteration for parametrization PI (—□—) and PII (—▲—) for ID = 7. Both optimization problems converge to a motion with significantly lower required total work; PII finds a better motion than PI (in terms of total work) due to the enlarged search space, at the cost of additional iterations. Each optimization iteration requires a primal flow computation – to evaluate the quantities of interest – and its corresponding adjoint – to evaluate the gradient of the quantity of interest. *Right*: Convergence of optimal value of total work $W$ as parameter space is refined for parametrization PI (—□—) and PII (—▲—). This implies convergence to an optimal, smooth trajectory that is not polluted by its discrete parametrization.

IDs. The first iteration corresponds to a steepest descent step, which causes an adverse jump in the objective value. The following iterations make rapid progress toward the optimal solution, which is slowed as convergence is approached. The solver requires additional iterations to converge the solution corresponding to parametrization PII, which is expected due to the larger parameter space.

Next, convergence of the total work as the parameter space is refined is considered in Fig. 7 (right) and Table 4. This implies the optimal trajectory among all twice continuously differentiable functions is being approached. For both parametrizations, the optimal value of the total work agrees to 3 digits between IDs 6 and 7 (roughly a factor of 2 difference in dimension of parameter spaces) and 2 digits between IDs 3 and 7 (roughly a factor of 10 difference in dimension of parameter spaces). This sweep of optimization problems was repeated with a low-order method ($p = 1$) with a comparable number of degrees of freedom ($N_{\boldsymbol{u}}$). As expected, the same local optimum was obtained and the convergence behavior with respect to refinement in $N_{\boldsymbol{\mu}}$ was very similar.

The motion of the airfoil and vorticity of the surrounding flow are shown in Fig. 8 (nominal trajectory), Fig. 9 (optimal solution under parametrization PI), and Fig. 10 (optimal solution under parametrization PII). The flow corresponding to the nominal configuration experiences flow separation and vortex shedding, which results in the relatively large amount of total energy to complete the mission. Fixing the translational motion and optimizing over the rotation (PI) dramatically reduces

**Table 4**
Table summarizing integrated quantities of interest at optimal solution of (56) for each parametrization (PI, PII) for each level of refinement. The total work monotonically increases as $N_\mu$ increases for a given parametrization, which is expected due to the nested search spaces. For a fixed ID, the optimal total work for parametrization PII is larger than that for PI since the search space for PI is a subset of that of PII. The other integrated quantities are included for completeness, but do not exhibit trends (except for converging to a fixed value as $N_\mu$ increases) since they were not included in the optimization problem.

| ID | PI | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $W$ | −2.1951e−01 | −1.5881e−01 | −1.5358e−01 | −1.5128e−01 | −1.5026e−01 | −1.4950e−01 | −1.4924e−01 |
| $W_x$ | 8.1329e−02 | 5.6090e−02 | 4.9543e−02 | 4.5924e−02 | 4.5085e−02 | 4.4712e−02 | 4.4707e−02 |
| $W_y$ | −2.3460e−01 | −1.8153e−01 | −1.7122e−01 | −1.6544e−01 | −1.6374e−01 | −1.6298e−01 | −1.6294e−01 |
| $W_\theta$ | 6.6234e−02 | 3.3370e−02 | 3.1906e−02 | 3.1768e−02 | 3.1604e−02 | 3.1223e−02 | 3.1010e−02 |
| $F_x$ | −1.9234e−01 | −1.3123e−01 | −1.1886e−01 | −1.1136e−01 | −1.0912e−01 | −1.0810e−01 | −1.0800e−01 |
| $F_y$ | −5.1539e−01 | −3.1711e−01 | −3.1816e−01 | −3.1877e−01 | −3.2551e−01 | −3.2959e−01 | −3.3063e−01 |

| ID | PII | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $W$ | −1.7357e−01 | −1.2095e−01 | −1.1733e−01 | −1.1629e−01 | −1.1603e−01 | −1.1557e−01 | −1.1502e−01 |
| $W_x$ | 9.6487e−03 | −1.4123e−02 | −1.4328e−02 | −1.4967e−02 | −1.5021e−02 | −1.5061e−02 | −1.5027e−02 |
| $W_y$ | −1.1041e−01 | −6.2238e−02 | −6.1036e−02 | −6.0425e−02 | −6.0032e−02 | −5.9489e−02 | −5.9245e−02 |
| $W_\theta$ | 7.2807e−02 | 4.4585e−02 | 4.1963e−02 | 4.0895e−02 | 4.0980e−02 | 4.1023e−02 | 4.0749e−02 |
| $F_x$ | −4.1265e−02 | 2.8091e−02 | 2.7677e−02 | 2.9596e−02 | 2.9848e−02 | 3.0231e−02 | 3.0221e−02 |
| $F_y$ | −3.2231e−01 | −1.064e−01 | −1.0806e−01 | −1.0890e−01 | −1.1343e−01 | −1.1626e−01 | −1.1764e−01 |

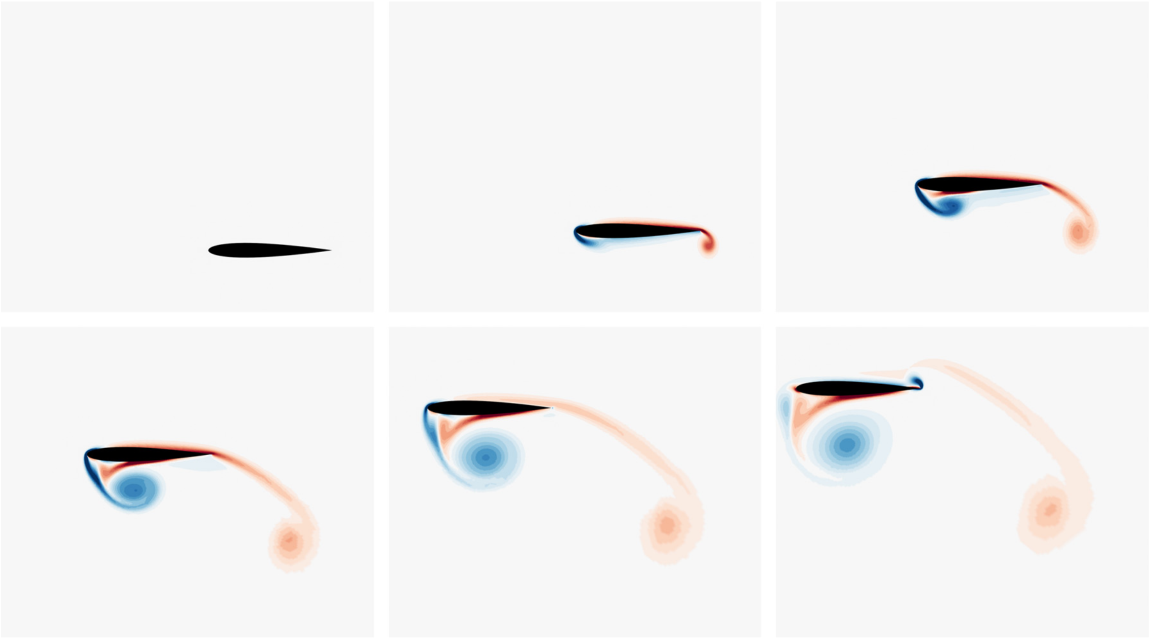

**Fig. 8.** Flow vorticity around airfoil undergoing motion corresponding to initial guess for optimization, i.e., pure heaving (—⊖—). Flow separation off leading edge implies a large amount of work required to complete mission. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

the amount of shedding and consequently reduces the amount of work required. Optimizing the entire rigid body motion (PII) further reduces the shedding and required work.

### 5.2. Energetically optimal shape and flapping motion of 2D airfoil at constant impulse

In this section, the high-order, time-dependent PDE-constrained optimization framework introduced in this document is applied to find the energetically optimal flapping motion, under an impulse constraint, of a 2D NACA0012 airfoil (Fig. 11) with chord length $l = 1$ and zero-thickness trailing edge. The governing equations are the 2D compressible, isentropic Navier–Stokes equations.

The goal is to determine the flapping motion – $y(t)$ and $\theta(t)$ – and shape – $c(t)$ – of the airfoil that minimizes the total energy such than a $x$-impulse of $q$ is achieved, i.e.,

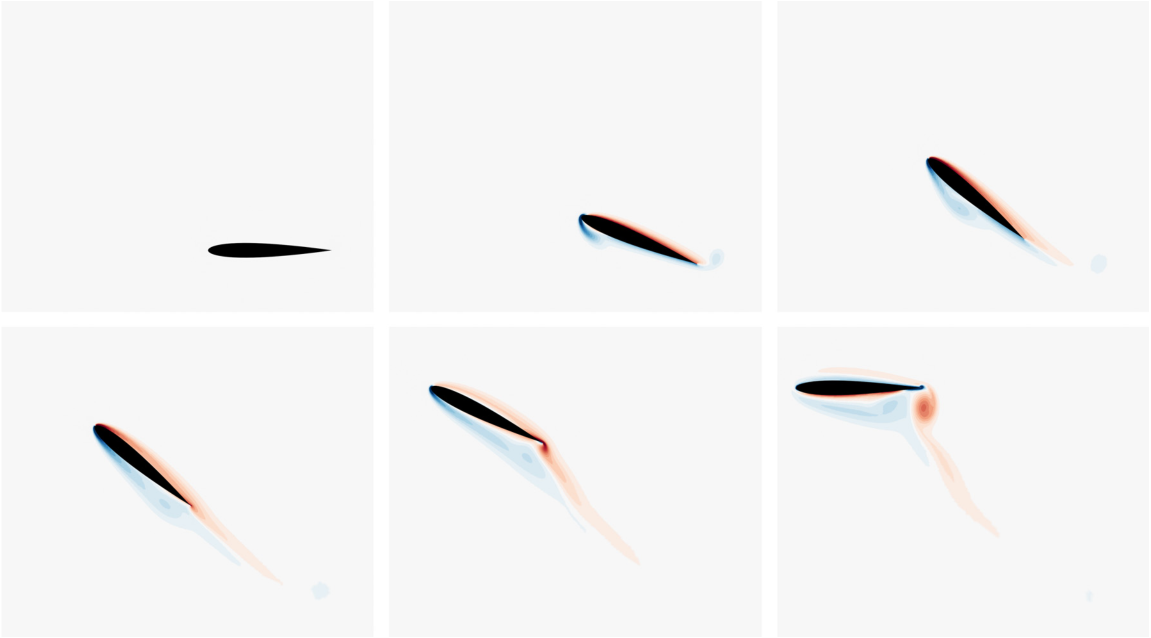

**Fig. 9.** Flow vorticity around airfoil undergoing motion corresponding to optimal pitching motion for fixed translational motion, i.e., solution of (56) under parametrization PI (—□—). The pitching motion greatly reduces the degree of flow separation and vortex shedding compared to the initial guess, and requires less work to complete the mission. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

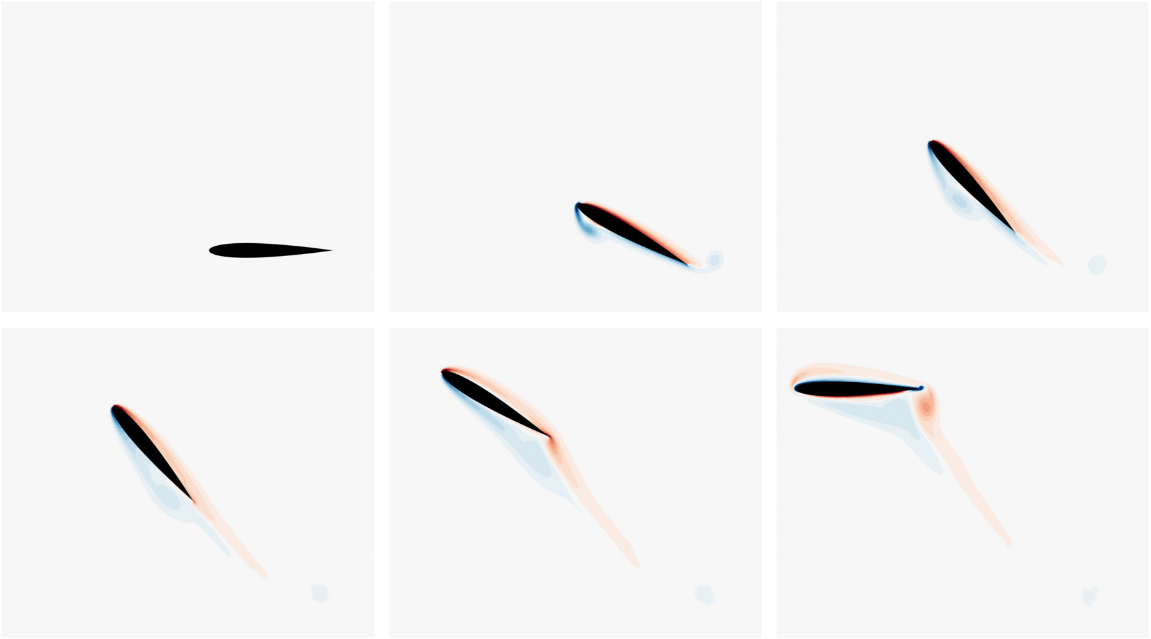

**Fig. 10.** Flow vorticity around airfoil undergoing motion corresponding to optimal rigid body motion, i.e. solution of (56) under parametrization PII (—△—). This rigid body motion further reduces the degree of flow separation and required work to complete the mission. This motion differs from the solution of PI as it has a larger pitch amplitude and slightly overshoots the final vertical position before settling to the required position. Snapshots taken at times $t = 0.0,\ 0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0$.

$$\begin{aligned} \underset{\boldsymbol{U},\,\boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{W}(\boldsymbol{U}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \mathcal{J}_x(\boldsymbol{U}, \boldsymbol{\mu}) = q \\ & \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U},\ \nabla \boldsymbol{U}) = 0 \quad \text{in } v(\boldsymbol{\mu}, t). \end{aligned} \tag{57}$$

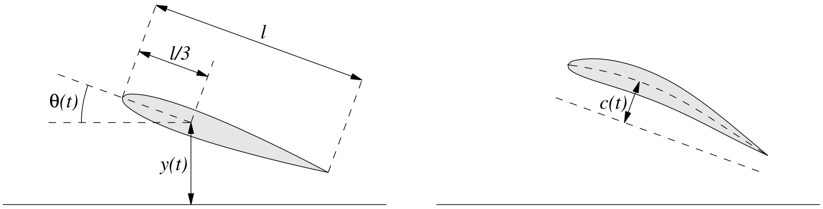

**Fig. 11.** Airfoil.

**Table 5**
Summary of parametrizations considered in Section 5.2. The number of periodic cubic spline knots used to discretize $y(t)$, $\theta(t)$, and $(t)$ are $m_y + 1$, $m_\theta + 1$, and $m_c + 1$, respectively. FI freezes the airfoil shape and considers only rigid body motions ($m_y = m_\theta \neq 0, m_c = 0$). FII parametrizes both shape and kinematic motion ($m_y = m_\theta = m_c \neq 0$).

| FI | | | | FII | | | |
|---|---|---|---|---|---|---|---|
| $m_y$ | $m_\theta$ | $m_c$ | $N_{\boldsymbol{\mu}}$ | $m_y$ | $m_\theta$ | $m_c$ | $N_{\boldsymbol{\mu}}$ |
| 4 | 4 | 0 | 6 | 4 | 4 | 4 | 9 |

The flapping frequency is fixed at 0.2, which corresponds to a period of $T = 5$. Proper initialization of the flow is the initial condition that results in a *time-periodic* flow [54] to completely avoid non-physical transients and simulate representative, *in-flight* conditions; this work uses a crude approximation that initializes the flow from the steady-state condition, simulates 3 periods of the flapping motion, and integrates the quantities of interest over the last period only. The deformation of the domain is determined from the value of $c(t)$ using the spatial blending map of Section 4.2 with

$$\boldsymbol{\varphi}(\boldsymbol{X}, \boldsymbol{\mu}, t) = \begin{bmatrix} 0 \\ 2c(t)e^{-[(\boldsymbol{X}-\boldsymbol{x}_0)\cdot \boldsymbol{e}_1]^2} \end{bmatrix}. \tag{58}$$

The deformation in (58) is volume preserving since $\det(\boldsymbol{I} + \nabla_{\boldsymbol{X}}\boldsymbol{\varphi}) \equiv 1$, but it does not preserve the surface area of the airfoil. A number of engineering situations exist where time-dependent shape changes alter the surface area of an object in a flow including control surface deployment on vehicles and extending panels on wings [55]. Additionally, morphing in nature is not necessarily surface area preserving [56]. Previous work that has considered optimal time-morphed geometries, using a vortex-lattice flow solver, also considered non-area-preserving deformations [57,58]. The time-dependent surface area is accounted for in the quantities of interest (54) through the spatial integral over the time-dependent boundary, $\boldsymbol{\Gamma}$, i.e., $\int_{\boldsymbol{\Gamma}} \cdot \, dS$.

The trajectory of the airfoil – $y(t)$, and $\theta(t)$ – and its shape – $c(t)$ – are discretized via cubic splines with $m_y + 1$, $m_\theta + 1$, and $m_c + 1$ knots, respectively, with boundary conditions that enforce

$$y(t) = -y(t + T/2) \qquad \theta(t) = -\theta(t + T/2) \qquad c(t) = -c(t + T/2). \tag{59}$$

These boundary conditions[2] for $y(t)$, $\theta(t)$, and $c(t)$ correspond to a mirroring of the trajectory at $t = T/2$ and implicitly enforces periodicity with period $T$. The knots are uniformly spaced between 0 and $T$ in the $t$-dimension and the knot values are optimization parameters. Since the unsteady simulation is initialized from the steady-state flow, non-zero velocities of the airfoil at $t = 0$ will result in non-physical transients. These transients are avoided by blending the periodic cubic spline smoothly to the zero function at the beginning of the time interval [22]. Let $s_y(t; \boldsymbol{\mu})$, $s_\theta(t; \boldsymbol{\mu})$, and $s_c(t; \boldsymbol{\mu})$ denote the periodic cubic spline approximations. Then, the flapping and shape trajectories are defined as

$$y(t) = b(t)s_y(t; \boldsymbol{\mu}) \qquad \theta(t) = b(t)s_\theta(t; \boldsymbol{\mu}) \qquad c(t) = b(t)s_c(t; \boldsymbol{\mu}), \tag{60}$$

where $b(t) = 1.0 - e^{-t^2}$. Table 5 summarizes two parametrizations considered in this section: (FI) rigid body motion parametrized via cubic splines and shape fixed at nominal value and (FII) rigid body motion and shape of airfoil parametrized via cubic splines. With this parametrization of the airfoil kinematics and shape, spatial and temporal discretization with the high-order scheme of Section 2 leads to the fully discrete version of the optimization problem in (57)

[2] Periodic and mirrored cubic splines of this form with $m + 1$ knots only have $m$ degrees of freedom since the boundary condition prescribes the value of the $m + 1$ knot from the values of the others $m$.

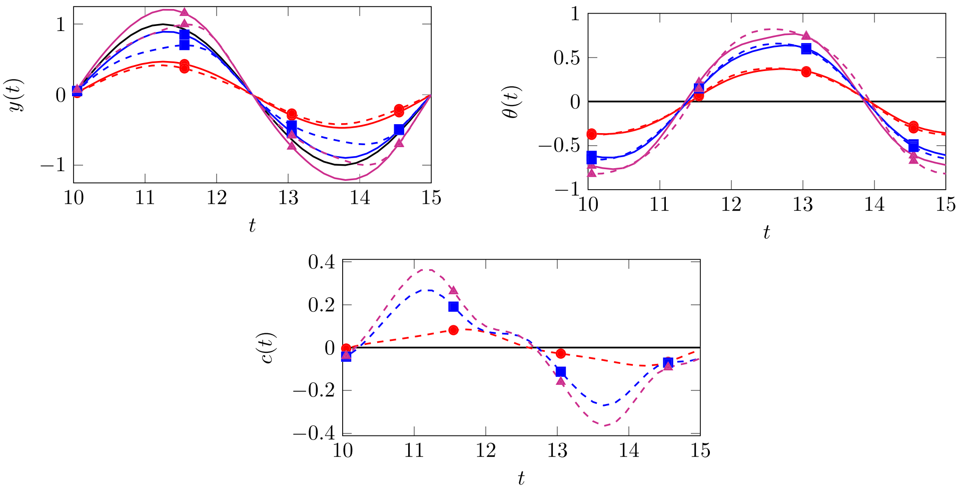

**Fig. 12.** Trajectories of $y(t)$, $\theta(t)$, and $c(t)$ at initial guess (—), solution of (61) under parametrization FI ($q = 0.0$: −●−, $q = 1.0$: −■−, $q = 2.5$: −▲−), and solution of (61) under parametrization FII ($q = 0.0$: - ● -, $q = 1.0$: - ■ -, $q = 2.5$: - ▲ - ) from Table 5.

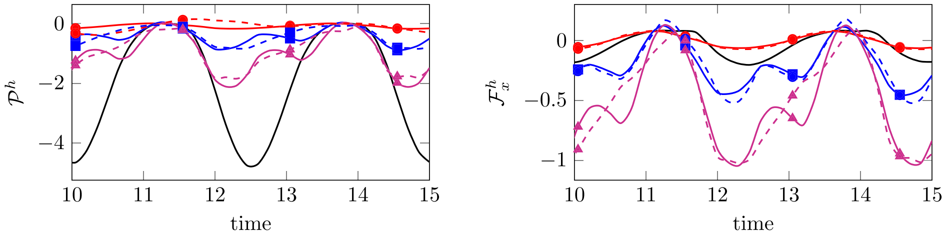

**Fig. 13.** Time history of total power, $\mathcal{P}^h(\boldsymbol{u}, \boldsymbol{\mu}, t)$, and $x$-directed force, $\mathcal{F}^h_x(\boldsymbol{u}, \boldsymbol{\mu}, t)$, imparted onto foil by fluid at initial guess (—), solution of (61) under parametrization FI ($q = 0.0$: −●−, $q = 1.0$: −■−, $q = 2.5$: −▲−), and solution of (61) under parametrization FII ($q = 0.0$: - ● -, $q = 1.0$: - ■ -, $q = 2.5$: - ▲ - ) from Table 5.

$$
\begin{aligned}
\underset{\substack{\boldsymbol{u}^{(0)},\,\dots,\,\boldsymbol{u}^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\,\dots,\,\boldsymbol{k}_s^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{\mu}\in\mathbb{R}^{N_{\boldsymbol{\mu}}}}}{\text{minimize}} \quad & W(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) \\
\text{subject to} \quad & J_x(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)},\ \boldsymbol{\mu}) = q \\
& \boldsymbol{u}^{(0)} = \boldsymbol{u}_0 \\
& \boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
& \mathbb{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right).
\end{aligned}
\tag{61}
$$

Given the gradient verification from the previous section, attention is turned directly to the optimization problem in (61) for various values of the impulse constraint, $q$. The optimization solver used in this section is SNOPT [59], a nonlinearly constrained SQP method. Fig. 12 contains the initial guess for the optimization problem in (56) as well as its solution under both parametrization, FI and FII. The initial guess for the optimization problem is a pure heaving motion at a fixed shape, i.e., $c(t) = \theta(t) = 0$. The solution under parametrization PI freezes the shape at its nominal configuration (NACA0012) and modifies the rigid body motion. Pitch is introduced for all values of the impulse constraint and the amplitude of the heaving motion is decreased for $q = 0.0, 1.0$ and increased for $q = 2.5$. The solution under parametrization PII reduces the heaving amplitude and slightly increases the pitch amplitude as compared to PI. It also introduces non-trivial camber.

The instantaneous quantities of interest – $W$ and $J_x$ in this case – for the nominal motion and shape and solution of (61) under parametrizations PI and PII are included in Fig. 13. It is clear that the optimal solution under both parametrizations result in a time history of the total power that is uniformly closer to 0 than that at the nominal trajectory, which is expected since $W$ is the objective function. It is also clear that larger values of the impulse constraint require more power to complete the flapping motion. While it may not be clear from Fig. 13, the integration of $\mathcal{F}^h_x$ leads to an impulse that exactly conforms to the specified value of $q$. This can be seen more clearly in Fig. 14. These observations can also be verified in Fig. 14 and Table 6 that summarizes the optimal values of the integrated quantities of interest.

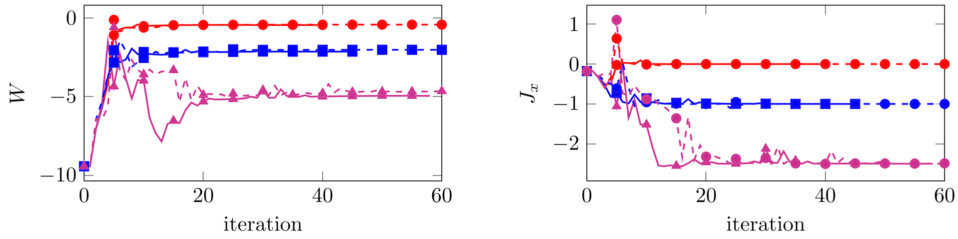

**Fig. 14.** Convergence of quantities of interest, $W$ and $J_x$, with optimization iteration for parametrization FI ($q = 0.0$: —●—, $q = 1.0$: —■—, $q = 2.5$: —▲—) and FII ($q = 0.0$: - ● -, $q = 1.0$: - ■ -, $q = 2.5$: - ▲ -) from Table 5. Each optimization iteration requires a primal flow computation – to evaluate quantities of interest – and its corresponding adjoint – to evaluate the gradient of quantities of interest.

**Table 6**
Table summarizing integrated quantities of interest at optimal solution of each optimization problem for each impulse level. In all cases, the desired value of $J_x$ is achieved to greater than 4 digits of accuracy. The optimal solution for larger values of the impulse constraint require more total work to complete flapping motion, i.e., work monotonically increases in magnitude as value of impulse constraint increases. Smaller values of total work are achievable if airfoil is allowed to morph its shape in addition its rigid body motion. The other integrated quantities are included for completeness, but do not exhibit trends since they were not in the optimization problem.

| $q$ | Initial | FI | | | FII | | |
|---|---|---|---|---|---|---|---|
| | | 0.0 | 1.0 | 2.5 | 0.0 | 1.0 | 2.5 |
| $W$ | −9.4096e+00 | −4.5695e−01 | −2.1419e+00 | −4.9476e+00 | −4.3252e−01 | −2.0271e+00 | −4.6110e+00 |
| $W_x$ | 0.0000e+00 | 0.0000e+00 | 0.0000e+00 | 0.0000e+00 | 0.0000e+00 | 0.0000e+00 | 0.0000e+00 |
| $W_y$ | −9.4096e+00 | −4.2807e−01 | −2.0642e+00 | −4.7967e+00 | −3.7363e−01 | −1.5413e+00 | −3.3712e+00 |
| $W_\theta$ | 0.0000e+00 | 2.8883e−02 | 7.7694e−02 | 1.5083e−01 | 1.7101e−02 | 6.7900e−03 | 1.8744e−01 |
| $F_x$ | −1.7660e−01 | −4.0490e−11 | −1.0000e+00 | −2.5000e+00 | 1.6937e−10 | −1.0000e+00 | −2.5000e+00 |
| $F_y$ | 3.5413e−01 | 1.5989e−02 | 5.0480e−02 | 9.7240e−02 | −1.5292e−02 | 4.8657e−02 | 9.6440e−02 |

Fig. 14 shows the convergence of the integrated quantities of interest with iterations in the optimization solver. The aforementioned observations can be verified by inspection of the final iteration: all impulse constraints are satisfied, larger values of $q$ require more work to achieve, and morphing the shape of the airfoil allows for a slight reduction in the required work. After 20 iterations, the impulse constraint is satisfied for $q = 0.0, 1.0$ and reduction of the work has essentially ceased, implying the optimization could have been terminated at that point. The case with $q = 2.5$ requires an additional 15–20 iterations to settle to a converged solution.

The shape and motion of the airfoil and vorticity of the surrounding flow are shown in Fig. 15 (nominal), Fig. 16 (optimal solution under parametrization FI for $q = 2.5$), and Fig. 17 (optimal solution under parametrization FII for $q = 2.5$). The flow corresponding to the nominal configuration experiences flow separation and vortex shedding, which results in the relatively large amount of total energy to complete the flapping motion and *does not satisfy the impulse constraint*. Fixing the shape and optimizing over the heaving and pitching motion (FI) dramatically reduces the amount of shedding and consequently reduces the amount of work required. Optimizing the shape in addition to the pitching and heaving motion (FII) further reduces the shedding and required work. The solution of FI and FII both satisfy the impulse constraint to greater than 8 digits of accuracy.

To conclude this section, a brief comparison of the optimal flapping motions found in this work are compared to those found in the literature. From Fig. 12, the pitch of the foil leads its plunge by approximately 90° in all optimal flapping motions, a result that was found in several works that range from experimental to computational [60,61,11,62]. The improved efficiency is largely due to a dramatic reduction in leading edge vortex shedding characteristic of pure heaving motions (Fig. 15) [60,11]. The specific pitching and heaving amplitudes were determined by the optimizer such that the impulse constraint is satisfied; as the impulse requirement is increased, the magnitude of the pitch and plunge increase and shedding off the leading edge is induced (Fig. 16) [62]. The time-dependent shape deformation slightly reduces the magnitude of the vortices shedding off the leading edge, which can be seen by comparing Figs. 16 and 17.

## 6. Conclusion

This document discussed an Arbitrary Lagrangian–Eulerian formulation of conservation laws on deforming domains and a high-order spatial discretization using a discontinuous Galerkin finite element method, previously introduced in [17]. A diagonally implicit Runge–Kutta scheme was used for high-order temporal integration with the benefit of achieving stability beyond second-order and not coupling all intermediate states like general, implicit Runge–Kutta methods. High-order spatial and temporal integration of quantities of interest was done in a solver-consistent manner, i.e., DG shape functions used for spatial integration and DIRK used for temporal integration. The result is a globally high-order accurate discretization of deforming domain conservation laws and relevant quantities of interest. The fully discrete adjoint equations for this globally high-order discretization were derived and the corresponding adjoint method introduced for computing gradients of quantities of interest along the manifold of high-order solutions of the conservation law. Implementation details regard-

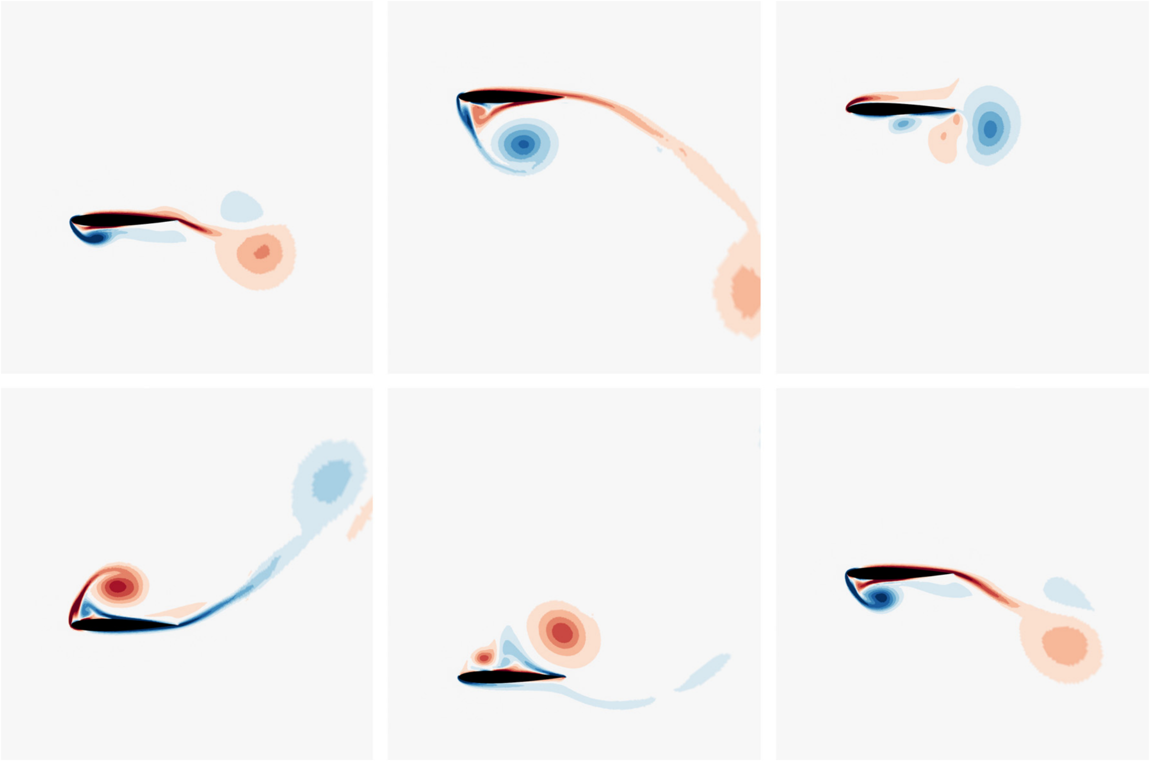

**Fig. 15.** Flow vorticity around flapping airfoil undergoing motion corresponding to initial guess for optimization problem (61), i.e., pure heaving (—). Flow separation off leading edge implies a large amount of work required for flapping motion. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

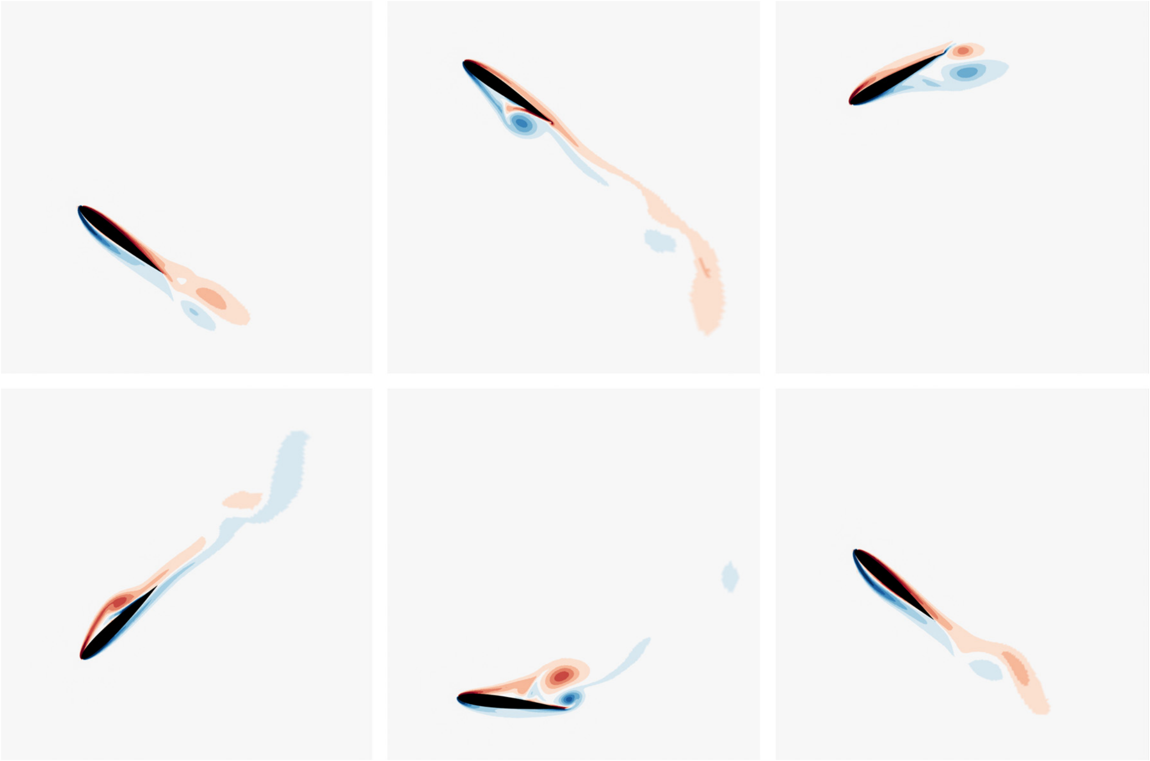

**Fig. 16.** Flow vorticity around flapping airfoil undergoing optimal rigid body motion corresponding to the solution of (61) under parametrization FI. The $x$-directed impulse is $J_x = 2.5$. The pitching motion greatly reduces the degree of flow separation and vortex shedding compared to the initial guess, and requires less work to complete the flapping motion and generate desired impulse. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

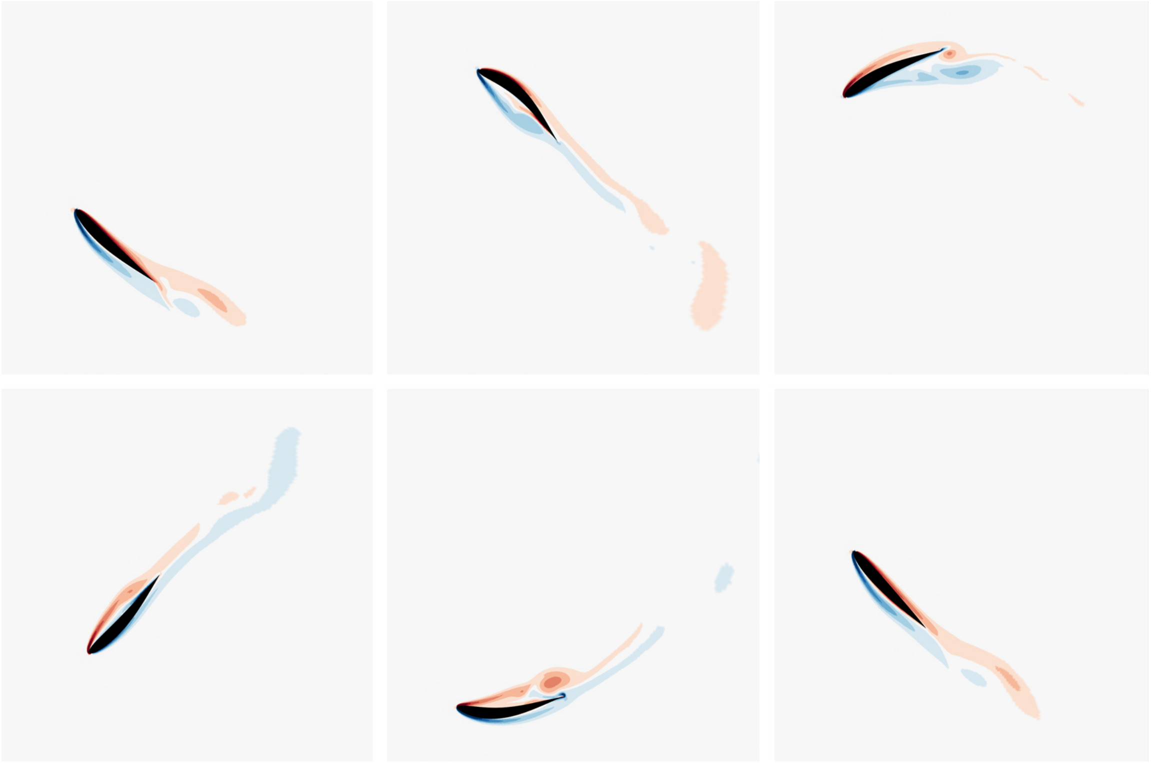

**Fig. 17.** Flow vorticity around flapping airfoil undergoing optimal deformation and kinematic motion, corresponding to the solution of (61) under parametrization FII. The $x$-directed impulse is $J_x = 2.5$. The morphing further reduces the flow separation and work required to complete the flapping motion and generate desired impulse. Snapshots taken at times $t = 9.75,\ 10.8,\ 11.85,\ 12.9,\ 13.95,\ 15.0$.

ing computation of necessary partial derivatives and domain deformation were included. The adjoint method was used in the context of gradient-based optimization to solve two optimal shape and control problems constrained by the isentropic, compressible Navier–Stokes equations, one of which incorporated a physics-based constraint. The first optimization problem was determination of the energetically optimal trajectory of an airfoil; the optimization solver reduced the energy required to complete the specified mission nearly an order of magnitude. The second optimization problem was the determination of the energetically optimal flapping motion and airfoil shape such that the $x$-directed impulse on the airfoil was fixed at a prescribed value; the solver found a configuration that satisfied the constraint to greater than 8 digits of precision and reduced the required energy.

This work is an initial step toward problems of engineering and scientific relevance. While the work extends to multi-physics problems defined on a single domain *without modification*, its extension to multiple domain problems – a canonical example being fluid–structure interaction problems – requires development, particularly if high-order accuracy is sought [51,63,25]. The extension to chaotic problems will also have significant engineering implications as this will be required to solve optimization problems governed by turbulent flow. Definition of a meaningful optimization problem in this context becomes a challenge as flow initialization [54] and sensitivity computation becomes difficult [64]. Low-order and high-order methods will both suffer from these difficulties as they arise from the chaotic nature of the physical problem. For this type of automated optimization approach to be competitive in engineering practice, the large cost associated with repeatedly solving the underlying partial differential equation must be addressed; two promising approaches are the use of inexact gradients [16] to accelerate adjoint computations and adaptive reduced-order models [48] to reduce the cost of the primal and sensitivity computations.

## Acknowledgements

This work was supported in part by the Department of Energy Computational Science Graduate Fellowship Program of the Office of Science and National Nuclear Security Administration in the Department of Energy under contract DE-FG02-97ER25308 (MZ), and by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under contract number DE-AC02-05CH11231 (PP). The content of this publication does not necessarily reflect the position or policy of any of these supporters, and no official endorsement should be inferred.

## Appendix A. Discrete adjoint equation derivation via Lagrange multiplier interpretation

In this section, an alternate derivation of the adjoint equations is provided using constrained optimization theory and the test variables introduced in Section 3.1 enter as *Lagrange multipliers* corresponding to constraints in an auxiliary optimization problem.

Fix $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{N_{\mu}}$ and consider the following *auxiliary optimization problem*, introduced purely for the derivation of the adjoint equations for a particular functional, $F$,

$$\begin{aligned} \underset{\substack{\boldsymbol{u}^{(0)},\,\dots,\,\boldsymbol{u}^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\,\dots,\,\boldsymbol{k}_s^{(N_t)}\in\mathbb{R}^{N_{\boldsymbol{u}}}}}{\text{minimize}} \quad & F(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)},\ \bar{\boldsymbol{\mu}}) \\ \text{subject to} \quad & \tilde{\boldsymbol{r}}^{(0)}(\boldsymbol{u}^{(0)},\ \bar{\boldsymbol{\mu}}) = 0 \\ & \tilde{\boldsymbol{r}}^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{u}^{(n)}, \boldsymbol{k}_1^{(n)}, \dots, \boldsymbol{k}_s^{(n)},\ \bar{\boldsymbol{\mu}}) = 0 \\ & \boldsymbol{R}_i^{(n)}(\boldsymbol{u}^{(n-1)},\ \boldsymbol{k}_1^{(n)}, \dots, \boldsymbol{k}_i^{(n)},\ \bar{\boldsymbol{\mu}}) = 0. \end{aligned} \tag{A.1}$$

Similar to Section 3.1, $F$ corresponds to any output functional whose gradient is desired. In the auxiliary optimization problem, *only the state vectors* are taken as optimization variables since $\bar{\boldsymbol{\mu}}$ is fixed. If the discrete PDE has a unique solution, the solution of (A.1) is equivalent to the solution of the fully discrete PDE in (24). In this case, the optimization problem (A.1) is simply a feasibility problem since the optimal solution is independent of the objective function.

The Lagrangian [65] corresponding to the optimization problem in (A.1) takes the form

$$\begin{aligned} &\mathcal{L}(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)}, \boldsymbol{\lambda}^{(0)},\ \dots,\ \boldsymbol{\lambda}^{(N_t)},\ \boldsymbol{\kappa}_1^{(1)},\ \dots,\ \boldsymbol{\kappa}_s^{(N_t)}) = \\ &\qquad F(\boldsymbol{u}^{(0)},\ \dots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \dots,\ \boldsymbol{k}_s^{(N_t)},\ \bar{\boldsymbol{\mu}}) - \sum_{n=0}^{N_t} {\boldsymbol{\lambda}^{(n)}}^T \tilde{\boldsymbol{r}}^{(n)} - \sum_{n=1}^{N_t}\sum_{i=1}^{s} {\boldsymbol{\kappa}_i^{(n)}}^T \boldsymbol{R}_i^{(n)}. \end{aligned} \tag{A.2}$$

Then, $(\boldsymbol{u}^{(0)*},\ \dots,\ \boldsymbol{u}^{(N_t)*},\ \boldsymbol{k}_1^{(1)*},\ \dots,\ \boldsymbol{k}_s^{(N_t)*}, \boldsymbol{\lambda}^{(0)*},\ \dots,\ \boldsymbol{\lambda}^{(N_t)*},\ \boldsymbol{\kappa}_1^{(1)*},\ \dots,\ \boldsymbol{\kappa}_s^{(N_t)*})$ is a critical point of (A.1) if it makes the Lagrangian stationary, i.e.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(n)}} &= 0 \qquad & n &= 0, \dots, N_t \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{k}_i^{(n)}} &= 0 \qquad & i &= 1, \dots, s,\ n = 1, \dots, N_t \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}^{(n)}} &= 0 \qquad & n &= 0, \dots, N_t \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\kappa}_i^{(n)}} &= 0 \qquad & i &= 1, \dots, s,\ n = 1, \dots, N_t. \end{aligned} \tag{A.3}$$

The equations in (A.3) are the first-order optimality conditions, or the Karush–Kuhn–Tucker (KKT) conditions. The conditions in (A.3) on the derivatives of the Lagrangian with respect to the Lagrange multipliers recover the constraints in (A.1). From (A.3), the condition $\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(N_t)}} = 0$ gives rise to

$${\frac{\partial \tilde{\boldsymbol{r}}^{(N_t)}}{\partial \boldsymbol{u}^{(N_t)}}}^T \boldsymbol{\lambda}^{(N_t)} = \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}. \tag{A.4}$$

The conditions on the derivatives of the Lagrangian with respect to the state variable $\boldsymbol{u}^{(n-1)}$ for $n = 1, \dots, N_t$, result in

$${\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}}^T \boldsymbol{\lambda}^{(n)} + {\frac{\partial \tilde{\boldsymbol{r}}^{(n-1)}}{\partial \boldsymbol{u}^{(n-1)}}}^T \boldsymbol{\lambda}^{(n-1)} = {\frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}}^T - \sum_{i=1}^{s} {\frac{\partial \boldsymbol{R}_i^{(n)}}{\partial \boldsymbol{u}^{(n-1)}}}^T \boldsymbol{\kappa}_i^{(n)}. \tag{A.5}$$

Finally, the conditions on the derivatives of the Lagrangian with respect to the stage variables, $\boldsymbol{k}_i^{(n)}$ for $n = 1, \dots, N_t$ and $i = 1, \dots, s$, are expanded as

$$\sum_{j=i}^{s} {\frac{\partial \boldsymbol{R}_j^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}}^T \boldsymbol{\kappa}_j^{(n)} = \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} - {\frac{\partial \tilde{\boldsymbol{r}}^{(n)}}{\partial \boldsymbol{k}_i^{(n)}}}^T \boldsymbol{\lambda}^{(n)}. \tag{A.6}$$

Equations (A.4)–(A.6), which correspond to the first two conditions in the KKT system (A.3) of the optimization problem in (A.1), are exactly the adjoint equations (29) derived in Section 3.1.

Substitution of the adjoint equation in (A.4)–(A.6) into the expression for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ in (28) recovers expression (30) for $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$, which is independent of the state sensitivities. Finally, substitution of the expressions in (24) for $\boldsymbol{r}^{(n)}$ and $\boldsymbol{R}_i^{(n)}$ in the adjoint equations (A.4)–(A.6) and gradient reconstruction formula (32) recover their specialization to the case of a DIRK temporal discretization, i.e. (31) and (32), respectively.

## References

[1] J.C. Newman III, A.C. Taylor III, R.W. Barnwell, P.A. Newman, G.J.-W. Hou, Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations, J. Aircr. 36 (1) (1999) 87–96.

[2] S. Nadarajah, A. Jameson, A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization, in: AIAA Paper, vol. 667, 2000, p. 2000.

[3] M.B. Giles, M.C. Duta, J.-D. Müller, N.A. Pierce, Algorithm developments for discrete adjoint methods, AIAA J. 41 (2) (2003) 198–205.

[4] D.J. Mavriplis, Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes, AIAA J. 45 (4) (2007) 741–750.

[5] C.A. Mader, J.R.A. Martins, J.J. Alonso, E.V. Der Weide, ADjoint: an approach for the rapid development of discrete adjoint solvers, AIAA J. 46 (4) (2008) 863–873.

[6] C.E. Orozco, O. Ghattas, Massively parallel aerodynamic shape optimization, Comput. Syst. Eng. 3 (1) (1992) 311–320.

[7] J. Reuther, J. Alonso, M.J. Rimlinger, A. Jameson, Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers, Comput. Fluids 28 (4) (1999) 675–700.

[8] O. Ghattas, J.-H. Bark, Optimal control of two- and three-dimensional incompressible Navier–Stokes flows, J. Comput. Phys. 136 (2) (1997) 231–244.

[9] B. He, O. Ghattas, J.F. Antaki, Computational strategies for shape optimization of time-dependent Navier–Stokes flows, TR-CMU-CML-97-102, Engineering Design Research Center, Carnegie Mellon Univ., 1997.

[10] N. Yamaleev, B. Diskin, E. Nielsen, Adjoint-based methodology for time-dependent optimization, in: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, 2008.

[11] M.F. Platzer, K.D. Jones, J. Young, J.S. Lai, Flapping wing aerodynamics: progress and challenges, AIAA J. 46 (9) (2008) 2136–2149.

[12] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, et al., High-order CFD methods: current status and perspective, Int. J. Numer. Methods Fluids 72 (8) (2013) 811–845.

[13] K. Mani, D.J. Mavriplis, Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes, AIAA J. 46 (2008) 1351–1364, 2015/06/22.

[14] M.J. Zahr, P.-O. Persson, Performance tuning of Newton–GMRES methods for discontinuous Galerkin discretizations of the Navier–Stokes equations, in: 21st AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2013.

[15] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, 1981.

[16] M. Heinkenschloss, L.N. Vicente, Analysis of inexact trust-region interior-point SQP algorithms, Department of Computational and Applied Mathematics, Rice University, 1995, TR95-18.

[17] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains, Comput. Methods Appl. Mech. Eng. 198 (17) (2009) 1585–1595.

[18] S.K. Nadarajah, A. Jameson, Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint method, AIAA J. 45 (7) (2007) 1478–1491.

[19] E.J. Nielsen, B. Diskin, N.K. Yamaleev, Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids, AIAA J. 48 (6) (2010) 1195–1206.

[20] K. Fidkowski, P. Roe, An entropy adjoint approach to mesh refinement, SIAM J. Sci. Comput. 32 (3) (2010) 1261–1287.

[21] K.J. Fidkowski, Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows, Int. J. Numer. Methods Eng. 88 (12) (2011) 1297–1322.

[22] M.P. van Schrojenstein Lantman, K. Fidkowski, Adjoint-based optimization of flapping kinematics in viscous flows, in: 21st AIAA Computational Fluid Dynamics Conference, 2013.

[23] S.M. Kast, K.J. Fidkowski, Output-based mesh adaptation for high order Navier–Stokes simulations on deformable domains, J. Comput. Phys. 252 (2013) 468–494.

[24] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff ODEs, SIAM J. Numer. Anal. 14 (6) (1977) 1006–1021.

[25] A. Mishra, K. Mani, D. Mavriplis, J. Sitaraman, Time dependent adjoint-based optimization for coupled fluid–structure problems, J. Comput. Phys. 292 (2015) 253–271.

[26] A. Sandu, On the properties of Runge–Kutta discrete adjoints, in: Computational Science—ICCS 2006, Springer, 2006, pp. 550–557.

[27] R. Joslin, M. Gunzburger, R. Nicolaides, G. Erlebacher, M. Hussaini, Self-contained automated methodology for optimal flow control, AIAA J. 35 (5) (1997) 816–824.

[28] W.W. Hager, Runge–Kutta methods in optimal control and the transformed adjoint system, 87 (2) (2000) 247–282.

[29] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2002) 1749–1779.

[30] R. Hartmann, Adjoint consistency analysis of discontinuous Galerkin discretizations, SIAM J. Numer. Anal. 45 (6) (2007) 2671–2696.

[31] R. Hartmann, T. Leicht, Generalized adjoint consistent treatment of wall boundary conditions for compressible flows, J. Comput. Phys. 300 (2015) 754–778.

[32] P. Houston, E. Süli, hp-Adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems, SIAM J. Sci. Comput. 23 (4) (2001) 1226–1252.

[33] K. Harriman, P. Houston, B. Senior, E. Süli, *hp*-Version discontinuous Galerkin methods with interior penalty for partial differential equations with nonnegative characteristic form, in: C.-W. Shu, T. Tang, S.-Y. Cheng (Eds.), Recent Advances in Scientific Computing and Partial Differential Equations, in: Contemporary Mathematics, vol. 330, AMS, 2003, pp. 89–119.

[34] K. Harriman, D. Gavaghan, E. Süli, The Importance of Adjoint Consistency in the Approximation of Linear Functionals using the Discontinuous Galerkin Finite Element Method, na-04/18, Oxford University Computing Laboratory, 2004.

[35] M.H. Imam, Three-dimensional shape optimization, Int. J. Numer. Methods Eng. 18 (5) (1982) 661–673.

[36] J.A. Samareh, A survey of shape parameterization techniques, in: NASA Conference Publication, 1999, pp. 333–344, Citeseer.

[37] C. Degand, C. Farhat, A three-dimensional torsional spring analogy method for unstructured dynamic meshes, Comput. Struct. 80 (3) (2002) 305–316.

[38] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids, J. Comput. Phys. 174 (2) (2001) 669–694.

[39] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.

[40] M. Jones, N.K. Yamaleev, Adjoint based shape and kinematics optimization of flapping wing propulsive efficiency, in: 43rd AIAA Fluid Dynamics Conference, San Diego, CA, 2013, 2013, AIAA 2013-2472.

[41] P.-O. Persson, Scalable parallel Newton–Krylov solvers for discontinuous Galerkin discretizations, in: 47th AIAA Aerospace Sciences Meeting, 2009, AIAA-2009-606.
[42] N.K. Yamaleev, B. Diskin, E.J. Nielsen, Local-in-time adjoint-based method for design optimization of unsteady flows, J. Comput. Phys. 229 (14) (2010) 5394–5407.
[43] I. Charpentier, Checkpointing schemes for adjoint codes: application to the meteorological model Meso-NH, SIAM J. Sci. Comput. 22 (6) (2001) 2135–2151.
[44] P. Heimbach, C. Hill, R. Giering, An efficient exact adjoint of the parallel MIT general circulation model, generated via automatic differentiation, Future Gener. Comput. Syst. 21 (8) (2005) 1356–1371.
[45] V. Heuveline, A. Walther, Online checkpointing for parallel adjoint computation in PDEs: application to goal-oriented adaptivity and flow control, in: Euro-Par 2006 Parallel Processing, Springer, 2006, pp. 689–699.
[46] J. Peraire, P.-O. Persson, The Compact Discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput. 30 (4) (2008) 1806–1824.
[47] M.J. Zahr, K. Carlberg, D. Amsallem, C. Farhat, Comparison of model reduction techniques on high-fidelity linear and nonlinear electrical, mechanical, and biological systems, Tech. rep., University of California, Berkeley, 2010.
[48] M.J. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, Int. J. Numer. Methods Eng. 102 (5) (2015) 1111–1135.
[49] C.-K. Lin, On the incompressible limit of the compressible Navier–Stokes equations, Commun. Partial Differ. Equ. 20 (3–4) (1995) 677–707.
[50] B. Desjardins, E. Grenier, P.-L. Lions, N. Masmoudi, Incompressible limit for solutions of the isentropic Navier–Stokes equations with Dirichlet boundary conditions, J. Math. Pures Appl. 78 (5) (1999) 461–471.
[51] B.M. Froehle, High-Order Discontinuous Galerkin Fluid–Structure Interaction Methods, University of California, Berkeley, 2013.
[52] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (2) (1981) 357–372.
[53] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560.
[54] M.J. Zahr, P.-O. Persson, J. Wilkening, A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints, Comput. Fluids: Special Issue on USNCCM13 International Symposium on Spectral and High-Order Methods (2016).
[55] A.H. Supekar, Design, analysis and development of a morphable wing structure for unmanned aerial vehicle performance augmentation, ProQuest, 2007.
[56] S. Ho, H. Nassef, N. Pornsinsirirak, Y.-C. Tai, C.-M. Ho, Unsteady aerodynamics and flow control for flapping wing flyers, Prog. Aerosp. Sci. 39 (8) (2003) 635–681.
[57] B.K. Stanford, P.S. Beran, Analytical sensitivity analysis of an unsteady vortex-lattice method for flapping-wing optimization, J. Aircr. 47 (2) (2010) 647–662.
[58] M. Ghommem, M.R. Hajj, D.T. Mook, B.K. Stanford, P.S. Beran, R.D. Snyder, L.T. Watson, Global optimization of actively morphing flapping wings, J. Fluids Struct. 33 (2012) 210–228.
[59] P.E. Gill, W. Murray, M.A. Saunders, SNOPT: an SQP algorithm for large-scale constrained optimization, SIAM J. Optim. 12 (4) (2002) 979–1006.
[60] I.H. Tuncer, M. Kaya, Optimization of flapping airfoils for maximum thrust and propulsive efficiency, AIAA J. 43 (11) (2005) 2329–2336.
[61] R. Ramamurti, W. Sandberg, Simulation of flow about flapping airfoils using finite element incompressible flow solver, AIAA J. 39 (2) (2001) 253–260.
[62] A. Oyama, Y. Okabe, K. Shimoyama, K. Fujii, Aerodynamic multiobjective design exploration of a flapping airfoil using a Navier–Stokes solver, J. Aerosp. Comput. Inf. Commun. 6 (3) (2009) 256–270.
[63] B. Froehle, P.-O. Persson, High-order accurate fluid–structure simulation of a tuning fork, Comput. Fluids 98 (2014) 230–238.
[64] Q. Wang, R. Hu, P. Blonigan, Least squares shadowing sensitivity analysis of chaotic limit cycle oscillations, J. Comput. Phys. 267 (2014) 210–224.
[65] J. Nocedal, S. Wright, Numerical Optimization, Series in Operations Research and Financial Engineering, Springer, 2006.